A Multi-scale Deep Learning Framework for Predicting Stratified Shear Flow Instabilities

Sangeetha Sankar,[1, 2, 3] Vallabh Deogaonkar,[1, 2, 4] Arjun Jagannathan,[2, 3] and Abhilash Somayajula[1, 2, 4]
[1)]*Marine Autonomous Vehicles (MAV) Laboratory, Indian Institute of Technology Madras, Chennai 600036, Tamil Nadu, India*
[2)]*Department of Ocean Engineering, Indian Institute of Technology Madras, Chennai 600036, Tamil Nadu, India*
[3)]*Centre of Excellence for Geophysical Flows Lab (GFL), Indian Institute of Technology Madras, Chennai 600036, Tamil Nadu, India*
[4)]*Centre of Excellence for Maritime Experiments to Maritime Experience (ME2ME), Indian Institute of Technology Madras, Chennai 600036, Tamil Nadu, India*

(*Electronic mail: [abhilash@iitm.ac.in])

(*Electronic mail: [arjunj@iitm.ac.in])

(Dated: 28 October 2026)

Kelvin–Helmholtz (KH) instability in stratified shear flows exhibits strong multiscale dynamics in which large-scale vortical structures evolve into smaller-scale motions through nonlinear interactions. Accurate prediction of KH dynamics is important for understanding mixing and transport in geophysical and environmental flows. Although high-fidelity simulations such as Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) can resolve these dynamics, their computational cost limits their use across a wide range of flow conditions. Data-driven reduced-order models based on Convolutional Neural Network Autoencoders (CNN-AEs) and Long Short-Term Memory (LSTM) networks offer a promising alternative but often underrepresent dynamically important fine-scale structures. This study presents a parameter-conditioned multiscale deep learning framework for reduced-order prediction of stratified KH instability. The framework employs a State CNN-AE to capture dominant large-scale flow structures, a Difference CNN-AE to learn finer-scale motions, and a parameter-conditioned LSTM to jointly evolve both latent representations using the gradient Richardson number (J) and the shear-to-stratification thickness ratio (R). The model is trained using vorticity and buoyancy fields from two-dimensional LES-like simulations spanning multiple flow configurations and is evaluated on both training and previously unseen test cases. The framework accurately reproduces the evolution of KH billows and preserves fine-scale structures throughout the nonlinear stages of instability development, as assessed through comparisons with the conventional CNN-AE–LSTM framework and LES using statistical metrics and physics-based diagnostics, including perturbation enstrophy, instability growth rates, and kinetic energy spectra. The proposed model also enables long-horizon forecasting from a single early-stage flow snapshot and generalizes successfully to previously unseen (J,R) combinations.

Keywords: Kelvin-Helmholtz instability, stratified shear flow, reduced-order modeling, convolutional autoencoder, long short-term memory network, multiscale deep learning

I. INTRODUCTION

Stratified shear instabilities occur when horizontal shear overcomes the stabilizing effect of vertical density stratification, leading to the formation of large-scale billows and eventually turbulence[1,2]. Among stratified shear instabilities, the Kelvin–Helmholtz (KH) instability is one of the most commonly observed and extensively studied. It occurs in a wide range of geophysical flows, including the oceans, lakes, estuaries, and the atmosphere. It plays a central role in the generation of turbulence, irreversible mixing, and vertical transport in stratified environments[3–6]. The resulting mixing redistributes momentum, heat, salinity, and other tracers, thereby influencing oceanic and atmospheric circulation. As KH billows grow and overturn, nonlinear interactions produce progressively smaller-scale flow structures, resulting in a cascade of kinetic energy from large scales to smaller scales. This process enhances mixing across density interfaces and generates complex multiscale flow structures[7,8]. Recent studies have further highlighted the intricate pathways through which fluid parcels are transported and mixed in stratified shear flows[9].

Accurately predicting the evolution of KH instability remains a major challenge in geophysical fluid dynamics. Large-scale coherent structures govern the overall flow evolution, whereas smaller-scale turbulent motions contribute significantly to mixing and dissipation. Capturing both behaviors is therefore essential for reliable prediction of stratified shear flows. Despite decades of research, understanding and quantifying KH-induced mixing in oceanic and atmospheric environments remain difficult problems, and the contribution of KH instability to mixing continues to be regarded as a first-order challenge[5,9,10].

The multiscale nature of KH instability poses significant computational challenges due to the wide range of spatial and temporal scales involved. High-fidelity methods such as Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES) can accurately resolve billow formation and turbulent breakdown, but they require substantial computational resources and are often impractical for exploring a broad range of background flow conditions[11]. The evolution and stability of KH flows depend on key nondimensional parameters, particularly the gradient Richardson number (Ri), the scale

ratio (R), the Reynolds number (Re), and the Prandtl number (Pr)[8,12]. For the hyperbolic tangent velocity and buoyancy profiles considered in this study, the centerline gradient Richardson number, $J = Ri(0)$, is adopted as the primary stability parameter. According to the Miles–Howard stability criterion, a gradient Richardson number exceeding $1/4$ everywhere is sufficient for stability; hence, $J = 1/4$ provides a classical critical bound for the present stratified shear-flow configuration[13,14]. Furthermore, the mixing behavior of stratified shear flows exhibits complex and often nonmonotonic dependence on these governing parameters[5,10,12]. These challenges have driven significant interest in developing more accurate and computationally efficient Reduced-Order Modeling (ROM) approaches that can capture the essential flow physics across a wide range of flow conditions at a fraction of the computational cost[15].

Recent advances in deep learning have enabled physics-informed approaches[16–19], as well as data-driven nonlinear ROM approaches that learn compact representations directly from high-dimensional flow data[20]. In particular, Convolutional Neural Network Autoencoders (CNN-AEs) provide an effective framework for extracting low-dimensional latent representations while preserving important spatial correlations and coherent structures present in the data[21]. Long Short-Term Memory (LSTM) networks can then be used to model the temporal evolution of these latent features for reduced-order prediction[21–23]. Such CNN-AE-LSTM frameworks have demonstrated promising predictive capabilities for complex fluid-flow systems and have shown potential for stratified-flow applications[12,24–26].

However, the multiscale nature of KH instability presents a significant challenge for data-driven reduced-order prediction, as the latent representation must simultaneously capture both the dominant coherent structures and the dynamically important fine-scale motions. Existing CNN-AE–LSTM frameworks typically rely on a single latent representation to describe the entire flow field[25,27]. Such representations may become biased toward the energetic large-scale features, leading to an underrepresentation of finer-scale structures. Consequently, accurately predicting the temporal evolution of the flow during the nonlinear stages of KH instability remains an open challenge.

Another important limitation is the ability to generalize across different background flow conditions. While most existing ML-ROMs do not explicitly incorporate governing flow parameters within the learning architecture, recent parameter-conditioned models have demonstrated improved predictive accuracy and robustness[23,28]. However, developing ROMs that simultaneously capture these multiscale features while remaining robust across different background flow conditions remains a significant challenge.

To address these limitations, we propose a scale-separated deep learning framework for reduced-order prediction of stratified shear flows and assess its performance using KH instability as a canonical benchmark problem. The key idea is that explicitly separating large-scale coherent structures from fine-scale mixing motions allows us to represent both scales using different latent variables, thereby more reliably capturing the underlying flow physics of KH evolution than conventional single-latent-space formulations.

In contrast to traditional CNN-AE–LSTM models that compress the entire flow into one latent representation[22,23,27], the present framework decomposes the flow into large and small-scale components and learns their motions separately. A State CNN-AE extracts a compact representation of the coherent structures governing the instability. A Difference CNN-AE with augmented inputs of the governing parameters J and R is proposed in this study to encode the finer-scale motions. The latent variables from these two encoders are then combined and advanced in time using a parameter-conditioned LSTM. By jointly forecasting the state and difference latent variables, the temporal model learns their coupled evolution, preserving information associated with both the initial instability and the subsequent nonlinear dynamics. This explicit multiscale latent-space design differs fundamentally from the existing approaches that rely on a single compressed representation of the flow.

To enhance generalization across different background flow conditions, the parameters J and R are embedded directly into the LSTM forecasting stage. This enables the temporal model to adapt its predictions to changes in stratification and scale ratio, allowing a single architecture to operate robustly over a wide range of governing flow parameters. In this way, the proposed framework combines explicit scale separation with parameter-aware temporal forecasting within a unified reduced-order modeling framework.

The present study investigates whether such a multiscale latent-space formulation can accurately reproduce the evolution of stratified KH instability over a range of (J,R) combinations. The proposed framework is evaluated for its ability to preserve multiscale flow dynamics while providing long-horizon forecasts from a single input snapshot across different background flow conditions. The framework is trained using high-fidelity two-dimensional (2D) LES data and tested on previously unseen cases. Evaluation of its performance through snapshot comparisons, statistical metrics and physics-based diagnostics shows consistently superior predictive capability compared to the traditional CNN-AE-LSTM framework.

The paper is organized as follows. Section II describes the numerical model and dataset generation procedure. Section III presents both the traditional and proposed deep-learning architectures, including the State CNN-AE, Difference CNN-AE, and parameter-conditioned LSTM network. Section IV discusses the results and discussion, where the predictive performance of the two frameworks is evaluated using statistical metrics and physics-based diagnostics. Finally, Section V summarizes the main findings and conclusions of the study.

II. NUMERICAL MODEL

The dataset used for training and evaluating the ROMs is generated from 2D LES-like numerical simulations. We say LES-like to emphasize the fact that although the simulations

are in 2D, they include a subgrid-scale closure model for turbulence. For brevity, we will henceforth refer to the simulations as LES. While three-dimensional (3D) effects become important in the development of secondary instabilities and eventual turbulent mixing at microscales[29], 2D KH instability retains the primary billow dynamics and key multiscale features of stratified shear flows, making it an attractive choice for ROM benchmarking. This section describes the governing equations, initial and boundary conditions, flow configuration, and simulation parameters used to generate the high-fidelity dataset. Simulations are conducted over a range of J and R to capture diverse instability behaviors and provide a comprehensive dataset for model training and evaluation.

A. Simulation Setup

Simulations are performed using the Julia-based Oceananigans.jl model[30], which solves the incompressible, nonhydrostatic Boussinesq equations using a finite-volume discretization. The model employs a fifth-order Weighted Essentially Non-Oscillatory (WENO) advection scheme combined with a third-order Runge-Kutta time integration. Subgrid-scale (SGS) turbulence is represented using the Smagorinsky-Lilly closure with a Smagorinsky coefficient $C_s = 0.16$[31,32], buoyancy coefficient $C_b = 1.0$[33,34], and turbulent Prandtl number $Pr = 1.0$[30]. The SGS closure introduces a spatially and temporally varying eddy viscosity that accounts for the effects of unresolved turbulent motions. Consequently, a fixed Reynolds number is not explicitly prescribed, and the effective dissipation characteristics of the flow are governed by the combined effects of the grid resolution and SGS model. The governing equations therefore correspond to the filtered incompressible, non-hydrostatic Boussinesq equations in an LES framework, where the influence of unresolved subgrid-scale motions is represented through the SGS closure. The equations are given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho_0}\nabla p + B\mathbf{k} + \nu_{\mathrm{mol}}\nabla^2\mathbf{u} - \nabla\cdot\boldsymbol{\tau}, \quad (1)$$

$$\frac{\partial B}{\partial t} + \mathbf{u}\cdot\nabla B = \kappa_{\mathrm{mol}}\nabla^2 B - \nabla\cdot\boldsymbol{\lambda}, \quad (2)$$

where $\mathbf{u} = (u, w)$ is the velocity vector, p is the pressure, $\mathbf{k}$ is the unit vector in the vertical direction and $B = -g\rho/\rho_0$ is the buoyancy, with ρ being the density and $\rho_0 = 1025\,\mathrm{kg\,m^{-3}}$ being a reference density. The parameters ν_{mol} and κ_{mol} denote the molecular viscosity and diffusivity, respectively. The tensors $\boldsymbol{\tau}$ and $\boldsymbol{\lambda}$ represent the subgrid-scale (SGS) stress tensor and buoyancy flux.

The computational domain is defined as $x \in [-10, 10]$ and $z \in [-5, 5]$, with uniform grid spacing $\Delta x = \Delta z = 0.078\,\mathrm{m}$. Periodic boundary conditions are imposed in the horizontal direction, such that $\phi(x + L_x, z, t) = \phi(x, z, t)$, for all prognostic variables $\phi \in \{u, w, B\}$, where L_x is the domain length in the horizontal direction. In the vertical direction, bounded boundary conditions are applied at $z = z_{\min}$ and $z = z_{\max}$, corresponding to impermeable and free-slip conditions for momentum and zero-flux conditions for buoyancy. These conditions are given by $w = 0$, $\frac{\partial u}{\partial z} = 0$, and $\frac{\partial B}{\partial z} = 0$, where u and w denote the horizontal and vertical velocity components, respectively, and B is the buoyancy field.

In the present study, broadband spatial white-noise perturbations are applied to the initial background flow fields shown in Fig. 1 to simulate realistic broadband disturbances and facilitate the development of multiscale flow structures. Although multiple modes are initially present, linear stability theory predicts that the fastest-growing mode dominates the early evolution. As the instability develops, nonlinear interactions govern its subsequent evolution. A fixed random seed is used for all simulations to ensure reproducibility of the white-noise perturbation field. The perturbations applied to the velocity and buoyancy fields are defined as the

$$u'(x,z) = \alpha\,\Delta U\,\mathscr{R}_u(x,z), \quad (3)$$

$$w'(x,z) = \alpha\,\Delta U\,\mathscr{R}_w(x,z), \quad (4)$$

$$B'(x,z) = \alpha\,\Delta B\,\mathscr{R}_B(x,z), \quad (5)$$

where u', w', and B' denote the perturbations to the horizontal velocity, vertical velocity, and buoyancy fields, respectively, while ΔU and ΔB denote the characteristic velocity and buoyancy differences across the shear and stratification layers, respectively. The quantities $\mathscr{R}_u$, $\mathscr{R}_w$, and $\mathscr{R}_B$ are Gaussian random fields with zero mean and unit variance. The parameters $\Delta U = U_{\max} - U_{\min}$ and $\Delta B = B_{\max} - B_{\min}$ denote the velocity and buoyancy differences across the shear and stratification layers, respectively, and $\alpha = 10^{-2}$ specifies the perturbation amplitude.

B. Flow Configuration

The canonical 2D KH instability configuration is employed to generate the high-fidelity dataset. The background velocity field, $\bar{\mathbf{U}}(z) = (\bar{U}(z), \bar{W}(z))$, and the background buoyancy field, $\bar{B}(z)$, are parameterized by the centerline gradient Richardson number, J, and the scale ratio, R. The initial background velocity field is prescribed using the hyperbolic tangent profile commonly adopted in stratified shear-layer studies[35],

$$\bar{\mathbf{U}}(z) = (\bar{U}(z), \bar{W}(z)) = (\tanh(z), 0), \quad (6)$$

which represents a free shear layer centered at $z = 0$.

To examine the effects of stable stratification, the background buoyancy field is prescribed as[35].

$$\bar{B}(z) = \frac{J}{R}\tanh(Rz), \quad (7)$$

The corresponding buoyancy frequency (N, also known as the Brunt–Väisälä frequency) is obtained from[35],

$$N^2(z) = \frac{d\bar{B}}{dz}, \quad (8)$$

where $R = h_u/h_b$ is a scale parameter defined as the ratio of shear layer thickness (h_u) to stratification layer thickness (h_b),

controls the vertical extent of the stratification layer. In this study, (h_u) is kept constant at unity, and the R is varied through h_b. The local gradient Richardson number is expressed as

$$Ri(z) = \frac{N^2(z)}{\left(\frac{d\bar{U}}{dz}\right)^2}, \tag{9}$$

where $J = Ri(0)$ denotes the centerline gradient Richardson number.

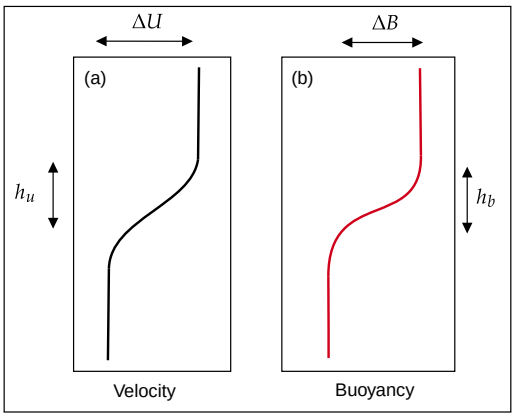


FIG. 1. Schematic representation of the background profiles. (a) Background velocity profile $\bar{U}(z)$; (b) Background buoyancy profile $\bar{B}(z)$. In this study, the velocity profile is fixed, while the buoyancy profile is varied through the parameters J and $R = h_u/h_b$, with J controlling the stratification strength and R controlling the relative thickness of the stratification layer.

The instability arises from the interaction between shear and stratification. The velocity shear extracts kinetic energy from the mean flow, promoting the growth of perturbations[9], whereas stratification suppresses vertical motion through buoyancy effects. The structure of the background profiles is shown in Fig. 1. In this configuration, the background velocity profile is held constant, while changes in the buoyancy profile through variations in J and R make it possible to investigate a range of stratification conditions[10]. For weak stratification (low J), the flow becomes unstable and is dominated by KH instability, characterized by the formation of large-scale coherent vortical structures.

C. Dataset Generation

The LES framework is used to generate datasets for training machine-learning (ML) models. Different flow conditions are obtained by varying the two nondimensional parameters of R and J[36]. A total of 32 cases are sampled uniformly across the (J,R) parameter space for training, with an additional three cases reserved for testing, as shown in Fig. 2. Each case corresponds to a unique combination of the governing parameters and therefore represents a distinct background stratification condition. The selected J values lie predominantly within the KH-instability regime ($J < 1/4$)[13,14].

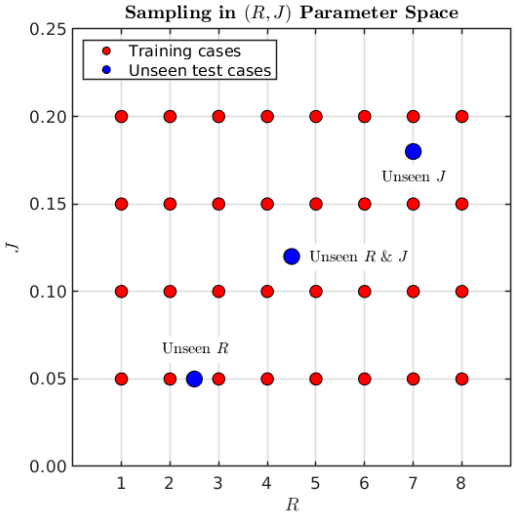


FIG. 2. Distribution of cases in the (J, R) parameter space. Each point represents a distinct initial background flow condition. Red markers denote training cases, while blue markers indicate unseen test cases used to evaluate model generalization.

TABLE I. Dataset configuration for ML model training and evaluation.

Parameter	Value
Simulation time per case	200 s
LES simulation time step	$\Delta t = 0.02$ s
ML data sampling interval	$\Delta t = 0.40$ s
Snapshots per simulation case	501
Training simulation cases	32
Unseen evaluation cases	3
Total training snapshots (32×501)	16,032
Training : Validation split	9:1
Training samples	14,428
Validation samples	1,604

The high-dimensional flow fields are generated using LES on a structured Cartesian grid comprising 256×128 grid points in the streamwise and vertical directions, respectively. All simulations employ a fixed time step of $\Delta t = 0.02$ s. A CFL sensitivity analysis was performed to determine the appropriate numerical time step, and the detailed analysis is provided in Appendix A. Each simulation is run up to $t = 200$ s, allowing the flow to evolve from the linear growth stage into the nonlinear regime. Flow-field snapshots are recorded every 0.40 s, yielding 501 snapshots per simulation case. The 32 training simulation cases therefore produce a total of 16,032 snapshots, which are randomly split into training and validation datasets using a 9:1 ratio. In addition, three unseen simulation cases are reserved exclusively for evaluating the generalization capability of the trained models. The overall dataset generation and data partitioning are summarized in Table I.

III. METHODOLOGY

The numerical simulations described above generate spatio-temporal datasets describing the evolution of the flow. The LES solves u, w, and B. Flow-field snapshots are recorded every $\Delta t = 0.40\,\mathrm{s}$, while the LES is advanced using a numerical time step of $\Delta t = 0.02\,\mathrm{s}$ over a total simulation time of 200 s. From u and w, the total vorticity Ω is computed as

$$\Omega = \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}. \tag{10}$$

The total buoyancy B is directly obtained from the LES. The velocity fields (u,w) can also be used to characterize the flow evolution. In the present study, Ω is selected as a derived flow representation because it directly emphasizes the vortical structures and velocity gradients associated with the KH instability. In addition, using (Ω,B) provides a more compact representation of the flow with two input fields rather than the three fields (u,w,B), thereby reducing the number of input fields used for ROM training. Both input fields are subsequently decomposed into background and perturbation components to separate the mean stratified shear-flow state from the instability-induced motions that develop during KH evolution.

The total fields are expressed as

$$\Omega(x,z,t) = \bar{\Omega}(z) + \Omega'(x,z,t), \tag{11}$$

and

$$B(x,z,t) = \bar{B}(z) + B'(x,z,t), \tag{12}$$

where $\bar{\Omega}(z)$ and $\bar{B}(z)$ denote the background vorticity and buoyancy fields, respectively, while $\Omega'(x,z,t)$ and $B'(x,z,t)$ represent the corresponding perturbation quantities. The background vorticity $\bar{\Omega}$ is computed from the prescribed background velocity profile (Eq. 6), while the perturbation vorticity is obtained from the perturbation velocity field Eqs. (3) and (4).

These flow field datasets serve as the input for training the ROM framework introduced in this study. The framework combines deep-learning-based dimensionality reduction with latent-space temporal forecasting to model the evolution of the flow. This section first presents a traditional approach consisting of a CNN-AE coupled with an LSTM network. Its performance is subsequently used as a reference for evaluating the proposed multiscale framework introduced in the following sections.

A. Traditional CNN-AE + LSTM Model

The CNN-AE + LSTM framework consists of two stages: (i) spatial compression of flow fields using a CNN-AE, and (ii) temporal prediction of the latent representations using an LSTM network. The predicted latent representations are then decoded to reconstruct the flow fields.

1. Convolutional Neural Network-based Autoencoder (CNN-AE)

The CNN-AE is employed to learn a compact low-dimensional representation of the high-dimensional flow fields while preserving the information required for accurate reconstruction[37]. As illustrated in Fig. 3, the encoder progressively compresses the input flow fields through successive convolution and pooling operations, and the decoder reconstructs the corresponding flow fields from the encoded representation. The convolutional architecture effectively extracts the spatial features associated with coherent flow structures.

Let $\mathbf{X}_{(i)} \in \mathbb{R}^{128\times256\times2}$ denote the input flow fields at the current snapshot (i), where the two channels correspond to Ω and B. The encoder, $\mathscr{E}$, compresses the input flow fields into a low-dimensional latent representation,

$$\mathbf{lv}_{(i)} = \mathscr{E}\left(\mathbf{X}_{(i)}\right), \qquad \mathbf{lv}_{(i)} \in \mathbb{R}^{4\times8\times16}. \tag{13}$$

The decoder, $\mathscr{D}$, reconstructs the flow snapshot from the latent representation, $\mathbf{lv}_{(i)}$, as

$$\hat{\mathbf{X}}_{(i)} = \mathscr{D}\left(\mathbf{lv}_{(i)}\right), \qquad \hat{\mathbf{X}}_{(i)} = \left(\hat{\Omega}_{(i)}, \hat{B}_{(i)}\right) \in \mathbb{R}^{128\times256\times2}, \tag{14}$$

where $\hat{\mathbf{X}}_{(i)}$ denotes the reconstructed flow snapshot, whose two channels correspond to the reconstructed vorticity, $\hat{\Omega}_{(i)}$, and reconstructed buoyancy, $\hat{B}_{(i)}$, respectively.

The encoder consists of five convolutional blocks, each comprising a convolutional layer with a 3×3 kernel, stride 1, padding 1, a tanh activation function, and a max-pooling layer of size 2. These operations progressively reduce the spatial resolution while extracting increasingly abstract flow features, resulting in the latent representation $\mathbf{lv}_{(i)}$, as summarized in Table II. The decoder $\mathscr{D}$ mirrors the encoder architecture using transposed convolution layers with a 2×2 kernel and stride 2 to reconstruct the flow fields.

The CNN-AE model is trained using a combined loss consisting of the mean squared error (MSE), ε_m, and a gradient difference loss, ε_g. The total loss, ε, is given by[23]

$$\varepsilon = \varepsilon_m + \varepsilon_g, \tag{15}$$

where ε_m enforces reconstruction accuracy and ε_g preserves gradient-based structural features. The MSE term is defined as

$$\varepsilon_m = \frac{1}{N_x N_y N_c} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_c} \left(q_{i,j,k} - \hat{q}_{i,j,k}\right)^2, \tag{16}$$

and the gradient difference loss is given by

$$\varepsilon_g = \frac{1}{N_c N_x N_y} \sum_{k=1}^{N_c} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left|(q_{i,j,k} - q_{i-1,j,k}) - (\hat{q}_{i,j,k} - \hat{q}_{i-1,j,k})\right| + \left|(q_{i,j,k} - q_{i,j-1,k}) - (\hat{q}_{i,j,k} - \hat{q}_{i,j-1,k})\right|, \tag{17}$$

where $q_{i,j,k}$ and $\hat{q}_{i,j,k}$ denote the ground truth and reconstructed values of the flow variables, respectively. The indices

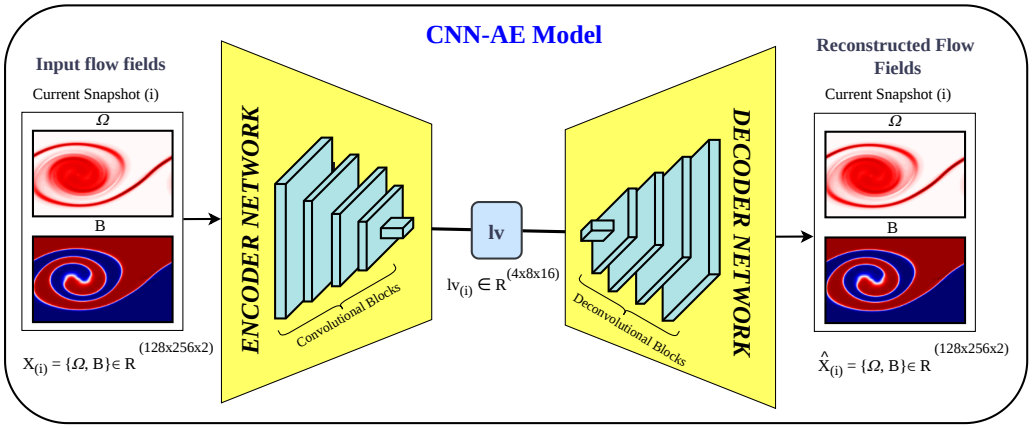


FIG. 3. Schematic of the CNN-AE architecture. The current input flow snapshot, $\mathbf{X}_{(i)} \in \mathbb{R}^{128\times256\times2}$, consists of two channels corresponding to $\Omega_{(i)}$ and $B_{(i)}$. The encoder comprises convolutional layers with 3×3 kernels, stride 1, tanh activation, and max-pooling of size 2, which progressively compress the input into the latent representation, $\mathbf{lv}_{(i)} \in \mathbb{R}^{4\times8\times16}$. The decoder reconstructs the corresponding flow snapshot, $\hat{\mathbf{X}}_{(i)} \in \mathbb{R}^{128\times256\times2}$, using transposed convolution layers. The detailed layer-wise architecture is summarized in Table II.

TABLE II. Architecture of the CNN-AE model. The input and output dimensions of each layer are denoted by (H,W,C), where H, W, and C represent the height, width, and number of channels, respectively. Each convolutional block consists of a convolutional layer followed by a tanh activation and max-pooling operation.

Encoder	Output Shape (H,W,C)	Decoder	Output Shape (H,W,C)
Input	$(128,256,2)$	Latent Representation	$(4,8,16)$
Conv Block 1	$(64,128,16)$	Transposed Conv Block 1	$(8,16,16)$
Conv Block 2	$(32,64,16)$	Transposed Conv Block 2	$(16,32,16)$
Conv Block 3	$(16,32,16)$	Transposed Conv Block 3	$(32,64,16)$
Conv Block 4	$(8,16,16)$	Transposed Conv Block 4	$(64,128,16)$
Conv Block 5	$(4,8,16)$	Transposed Conv Block 5	$(128,256,2)$
Final Conv (1×1)	$(4,8,16)$	Output	$(128,256,2)$

i and j correspond to the spatial grid locations in the x and y directions, while k represents the channel index (Ω and B). The quantities N_x and N_y denote the number of grid points in the spatial directions, and N_c is the total number of channels. The encoder and decoder networks in the CNN-AE model are optimized using hyperparameters listed in Table III. Once the model is trained, the encoder projects flow snapshots into latent representations, which serve as inputs to the temporal prediction model described next.

2. *Long Short-Term Memory (LSTM)*

The temporal evolution of the latent representations extracted from the CNN-AE is propagated forward through the LSTM network. The LSTM learns the nonlinear dynamics of the latent representations $\mathbf{lv}_{(i)} \in \mathbb{R}^{4\times8\times16}$, where (i) denotes the temporal index of the flow snapshot. The model is trained to perform one-step prediction in the latent space. The latent representation is flattened into a latent vector, denoted by

$$\mathbf{Z}_{(i)} \in \mathbb{R}^{512}. \tag{18}$$

TABLE III. Optimized hyperparameters used in the CNN-AE model.

Hyperparameter	Selected Value
Number of convolutional layers	10
Number of filters per layer	16
Kernel size	3×3
Activation function	tanh
Pooling operation	Max pooling
Upsampling operation	Transposed convolution
Loss function	ε refer to Eq. (15)
Loss weight $(\Omega:B)$	1 : 1
Optimizer	Adam[38]
Learning rate	10^{-3}
Batch size	80
Number of epochs	8000

The governing flow parameters J and R, are augmented with $\mathbf{Z}_{(i)}$ to form the LSTM input vector

$$\mathbf{h}_{(i)} = \left[\mathbf{Z}_{(i)}, J, R\right], \qquad \mathbf{h}_{(i)} \in \mathbb{R}^{514}. \tag{19}$$

The augmented input vector $\mathbf{h}_{(i)}$ is passed through the stacked LSTM layers to model the temporal evolution of the

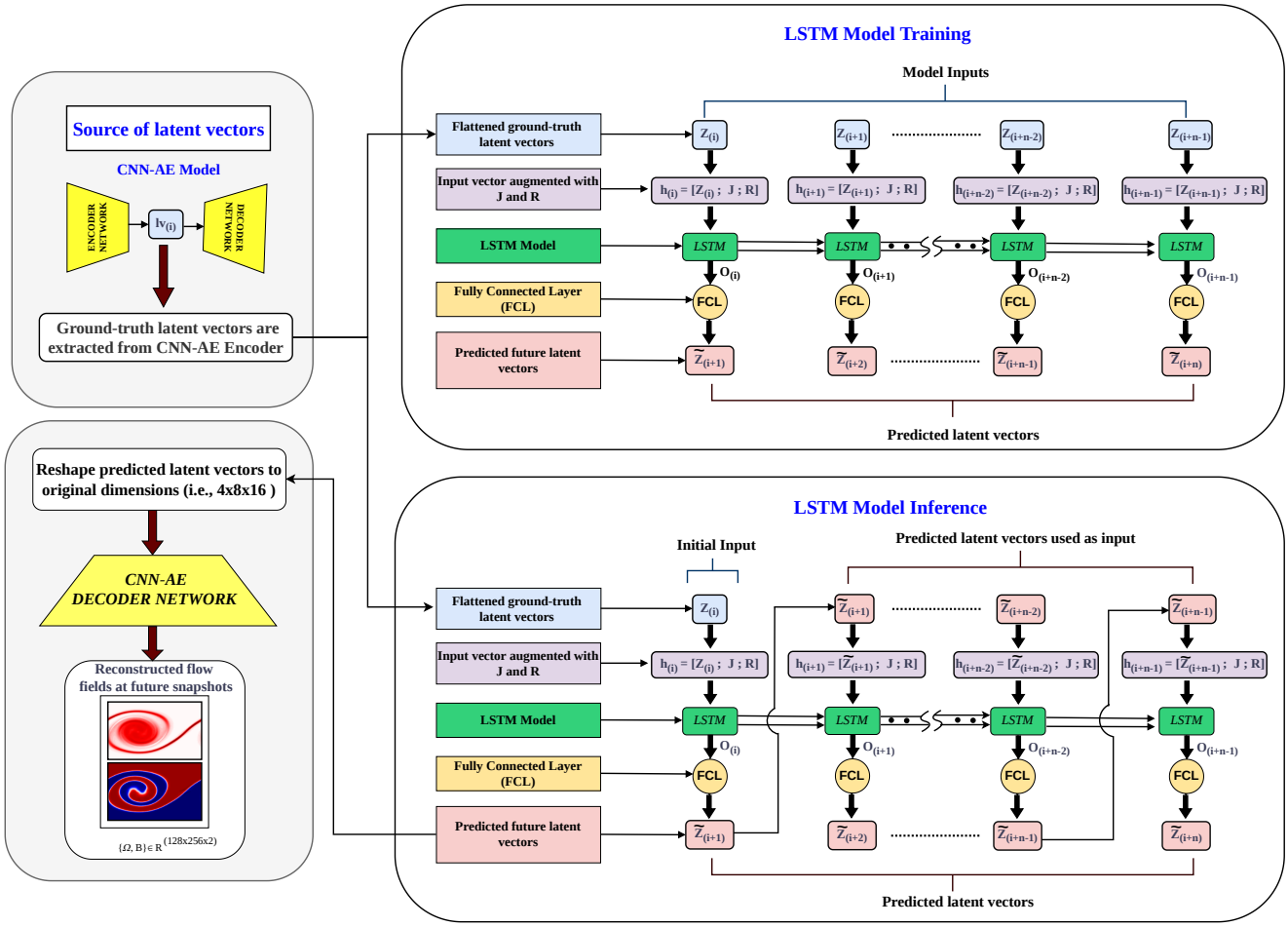


FIG. 4. Schematic of the parameter-conditioned LSTM framework used for latent-space forecasting. Ground-truth latent representations extracted from the CNN-AE encoder are first flattened into latent vectors, $\mathbf{Z}_{(i)} \in \mathbb{R}^{512}$, which are augmented with the governing flow parameters, J and R, to form the parameter-conditioned input vectors, $\mathbf{h}_{(i)} \in \mathbb{R}^{514}$. During training, the LSTM is trained using a one-step prediction framework, and the output of the final LSTM layer is projected through a FCL to predict the latent vector at the subsequent time step, $\widetilde{\mathbf{Z}}_{(i+1)}$, corresponding to a prediction interval of $\Delta t = 0.40\,\mathrm{s}$. During inference, the predicted latent vector is recursively augmented with J and R and supplied back to the LSTM to generate long-horizon latent-space predictions over the entire simulation duration (200 s). Finally, the predicted latent vectors are reshaped to their original dimensions and passed through the CNN-AE decoder to reconstruct the corresponding future vorticity and buoyancy fields, Ω and B.

latent representation while accounting for variations in the governing flow parameters. The output of the final LSTM layer, denoted by $\mathbf{o}_{(i)}$, is given by

$$\mathbf{o}_{(i)} = \mathscr{L}\left(\mathbf{h}_{(i)}\right), \qquad \mathbf{o}_{(i)} \in \mathbb{R}^{512}, \tag{20}$$

where $\mathscr{L}(\cdot)$ denotes the nonlinear mapping performed by the LSTM. The output of the final LSTM layer is subsequently projected onto the latent space through a fully connected output layer,

$$\widetilde{\mathbf{Z}}_{(i+1)} = \mathbf{W}_{fc}\mathbf{o}_{(i)} + \mathbf{b}_{fc}, \qquad \widetilde{\mathbf{Z}}_{(i+1)} \in \mathbb{R}^{512}, \tag{21}$$

where $\mathbf{W}_{fc}$ and $\mathbf{b}_{fc}$ denote the learnable weight matrix and bias vector of the fully connected layer (FCL). The predicted latent vector, $\widetilde{\mathbf{Z}}_{(i+1)}$, is then augmented with the governing flow parameters to construct the input vector, $\mathbf{h}_{(i+1)}$, for the subsequent prediction,

$$\mathbf{h}_{(i+1)} = \left[\widetilde{\mathbf{Z}}_{(i+1)}, J, R\right], \qquad \mathbf{h}_{(i+1)} \in \mathbb{R}^{514}, \tag{22}$$

During training, the LSTM is trained to perform one-step prediction using the parameter-conditioned input vector, $\mathbf{h}_{(i)}$, constructed from the latent vector, $\mathbf{Z}_{(i)}$, and the governing flow parameters, J and R, to predict the latent vector at the subsequent time step, $\widetilde{\mathbf{Z}}_{(i+1)}$, as described by Eqs. (19)–(22). The predicted latent vector is compared with the corresponding ground-truth latent vector, $\mathbf{Z}_{(i+1)}$, through the loss function defined in Eq. (23). During inference, the trained LSTM is used in a recursive prediction manner, where the predicted latent vector, $\widetilde{\mathbf{Z}}_{(i+1)}$, is augmented with the governing flow parameters to construct the input vector, $\mathbf{h}_{(i+1)}$, for the subse-

quent prediction. This recursive procedure is repeated at every prediction interval of $\Delta t = 0.40\,\mathrm{s}$ until the desired prediction horizon is reached. In the present study, recursive predictions are performed over the entire simulation duration of 200 s for a given flow condition specified by J and R. The predicted latent vectors are subsequently reshaped to their original dimensions and passed through the CNN-AE decoder to reconstruct the corresponding Ω and B fields, as illustrated in Fig. 4.

The LSTM architecture consists of three stacked recurrent layers with 512 hidden units per layer and a dropout rate of 0.2 applied between successive layers. The output of the final LSTM layer is passed through a fully connected output layer to obtain the predicted latent vector, which is subsequently reshaped to the original latent-space dimensions and provided as input to the decoder for flow-field reconstruction.

The LSTM model is trained by minimizing the mean squared error (MSE) between the predicted and ground-truth latent vectors,

$$\varepsilon_m = \frac{1}{N_s}\sum_{k=1}^{N_s}\left\|\widetilde{\mathbf{Z}}_k - \mathbf{Z}_k\right\|^2, \tag{23}$$

where ε_m denotes the mean squared error loss, $\widetilde{\mathbf{Z}}_k$ and $\mathbf{Z}_k$ denote the predicted and ground-truth latent vectors for the kth training sample, respectively, and N_s denotes the number of training samples in a batch. The optimized hyperparameters used for the LSTM model are summarized in Table IV.

TABLE IV. Optimized hyperparameters used in the LSTM model.

Hyperparameter	Selected Value
Input dimension	514
Hidden dimension	512
Output dimension	512
Number of LSTM layers	3
Dropout rate	0.2
Fully connected output layer	Yes
Loss function	ε_m refer to Eq. (23)
Optimizer	Adam[38]
Learning rate	10^{-3}
Batch size	80
Number of epochs	3000

3. *Limitations of the Traditional CNN-AE + LSTM Model*

Preliminary assessments of the traditional CNN-AE + LSTM model revealed several limitations in predicting the temporal evolution of KH instability. Although the framework accurately reconstructs the dominant large-scale flow structures, its performance deteriorates in representing the small-scale flow motions that emerge during the nonlinear stages of the instability. Since the CNN-AE compresses the entire flow field into a single latent representation, the dominant large-scale structures contribute more significantly to the reconstruction loss and tend to overshadow the small-scale motions. Consequently, the latent representation primarily captures the dominant flow dynamics, while the finer-scale structures are represented with reduced accuracy.

In addition, the spatial encoder is not explicitly conditioned on the governing flow parameters, limiting its ability to represent parameter-dependent flow features across different combinations of J and R values. Preliminary analysis also revealed phase discrepancies in the predicted temporal evolution during the nonlinear stages of the instability, indicating reduced accuracy in tracking the evolution of coherent flow structures, as observed, for example, in the testing case at $t = 100$s (Fig. 7).

These observations motivate the development of a scale-separated (multiscale) framework that independently represents the dominant large-scale and small-scale flow motions while incorporating parameter-dependent information. Detailed quantitative evidence supporting these observations is presented in Section IV.

B. Proposed Scale-Separated Framework: Multiscale CNN-AE + LSTM Model

The proposed multiscale framework is motivated by the limitations of the traditional CNN-AE + LSTM model discussed in Section III A 3. The underlying idea is that a single latent representation primarily focuses on dominant large-scale flow structures because they contribute more to the reconstruction error than small-scale motions.

As a result, the model tends to capture large-scale dynamics well but underrepresents the smaller features. To overcome this, the proposed framework decomposes the flow into large- and small-scale components and learns a dedicated latent representation for each component. Both these latent representations are concatenated and their temporal evolution is jointly predicted through the LSTM.

The proposed multiscale architecture consists of three interconnected networks (Networks A–C), as illustrated in Fig. 5. Each network is designed to model a distinct aspect of the flow. Network A, referred to as the State CNN-AE, extracts the latent representation of the dominant large-scale flow structures. Network B, referred to as the Difference CNN-AE, learns the latent representation of the small-scale flow motions. The latent representations obtained from Networks A and B are subsequently combined and augmented with the governing flow parameters, J and R, before being supplied to Network C. The parameter-conditioned LSTM then jointly learns the temporal evolution of the combined latent representations across different background flow conditions.

Thus, the large- and small-scale components are separated only at the spatial representation stage, while their temporal evolution is jointly propagated forward in time through the parameter-conditioned LSTM. This allows the proposed framework to retain information from both large- and small-scale structures while learning their combined temporal evolution during the development of the KH instability. The following subsections describe the three networks and their respective roles within the proposed framework.

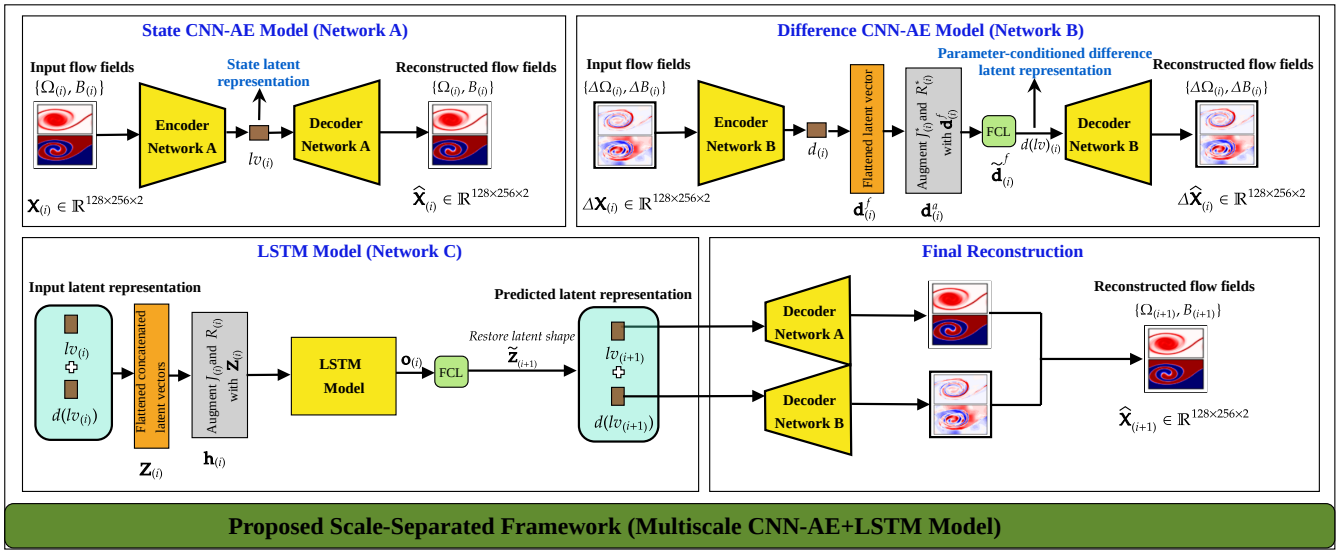


FIG. 5. Schematic of the proposed multiscale CNN-AE + LSTM framework. The input flow snapshot is first processed by Network A (State CNN-AE), which extracts the latent representation of the dominant large-scale flow structures and reconstructs the corresponding large-scale flow fields. The difference between the original flow snapshot and the reconstruction from Network A is subsequently supplied to Network B (Difference CNN-AE), where the governing flow parameters, J^* and R^*, are incorporated through latent-space parameter augmentation to learn the latent representation of the small-scale flow motions. The latent representations obtained from Networks A and B are concatenated and augmented with the governing flow parameters, J and R, to form the input to Network C (LSTM), which learns their coupled temporal evolution in the latent space. The predicted latent representations are subsequently decoded through the respective decoders of Networks A and B to reconstruct the large- and small-scale flow components, which are finally combined to obtain the predicted flow snapshot. During inference, the predicted latent representation is recursively supplied to the LSTM to enable long-horizon prediction. The detailed mathematical formulation of each network is presented in the corresponding methodology sections.

1. *State CNN-AE Model: Network A*

Network A is identical to the traditional CNN-AE model described in Section III A 1 and illustrated in Fig. 3. The same network architecture and optimized hyperparameters are retained in the proposed framework to ensure a direct and fair comparison with the traditional CNN-AE model. In the proposed multiscale framework, Network A, referred to as the *State CNN-AE*, learns a low-dimensional latent representation of the dominant large-scale flow structures, referred to as the *state* latent representation. To capture the small-scale flow motions, a second network, referred to as the Difference CNN-AE, is introduced and described in the following subsection.

2. *Difference CNN-AE Model: Network B*

Network B is designed to learn the small-scale flow motions that are not captured by Network A, referred to as the *Difference CNN-AE* model. Let $\hat{\mathbf{X}}^A_{(i)}$ denote the reconstructed flow snapshot obtained from Network A. The input to Network B, denoted by $\Delta\mathbf{X}_{(i)}$, is defined as the difference between the original flow snapshot and its corresponding reconstruction from Network A at current snapshot i,

$$\Delta\mathbf{X}_{(i)} = \mathbf{X}_{(i)} - \hat{\mathbf{X}}^A_{(i)} = \left(\Omega_{(i)} - \hat{\Omega}^A_{(i)},\, B_{(i)} - \hat{B}^A_{(i)}\right), \tag{24}$$

where

$$\Delta\mathbf{X}_{(i)} = \left(\Delta\Omega_{(i)}, \Delta B_{(i)}\right) \in \mathbb{R}^{128\times256\times2}, \tag{25}$$

denotes the difference flow fields of vorticity ($\Delta\Omega_{(i)}$) and buoyancy ($\Delta B_{(i)}$). These difference flow fields predominantly contain the small-scale flow motions that are not captured by the State CNN-AE and therefore serve as the input to Network B.

Before training Network B, the input difference flow fields are normalized using maximum absolute value scaling. The normalized difference vorticity ($\Delta\Omega^*_{(i)}$) and buoyancy fields ($\Delta B^*_{(i)}$) are given by

$$\Delta\Omega^*_{(i)} = \frac{\Delta\Omega_{(i)}}{\Delta\Omega_{\max}}, \qquad \Delta B^*_{(i)} = \frac{\Delta B_{(i)}}{\Delta B_{\max}}, \tag{26}$$

where $\Delta\Omega_{\max}$ and $\Delta B_{\max}$ denote the maximum absolute values of the corresponding difference fields computed over the entire training dataset. This normalization scales the input variables to comparable numerical magnitudes (approximately within the range $[-1, 1]$), thereby preventing variables with larger values from dominating the optimization process.

Since Network B employs the same encoder–decoder architecture as Network A, its schematic representation is omitted for brevity. Apart from the incorporation of the governing flow parameters in the latent space, Network B retains the same architecture and training configuration as the State

CNN-AE (Tables II and III). The principal modifications include increasing the number of convolutional filters from 16 to 32 to improve the representation of fine-scale flow motions, reducing the learning rate to 10^{-4}, and extending the training duration to 15 000 epochs. The effect of increasing the number of convolutional filters is examined through an ablation study in Appendix B, which supports the choice of 32 filters used in this study.

The encoder, $\mathscr{E}$, compresses the normalized difference flow fields into a latent representation, hereafter referred to as the *difference latent representation*,

$$\mathbf{d}_{(i)} = \mathscr{E}\left(\Delta\mathbf{X}^*_{(i)}\right), \qquad \mathbf{d}_{(i)} \in \mathbb{R}^{4\times8\times32}, \tag{27}$$

where $\mathbf{d}_{(i)}$ denotes the latent representation of the normalized difference flow fields.

In the initial implementation of Network B, the governing flow parameters J and R were not incorporated, consistent with the architecture of Network A. Since Network A successfully learns a low-dimensional representation of the dominant large-scale flow structures directly from the flow snapshots, explicit conditioning on the governing flow parameters was not initially considered necessary. However, preliminary experiments during model development showed that this formulation resulted in reduced reconstruction accuracy of the difference flow fields and poorer generalization across different stratification conditions.

Unlike the dominant large-scale structures learned by Network A, the difference input flow fields, $\Delta\mathbf{X}_{(i)}$, in Network B predominantly contain small-scale flow motions whose spatial distribution and temporal evolution depend strongly on the governing flow parameters, J and R. Consequently, the information contained in the difference flow fields alone is insufficient to consistently represent these parameter-dependent small-scale dynamics. To overcome this limitation, the normalized governing flow parameters, $J^*_{(i)}$ and $R^*_{(i)}$, are incorporated into the latent representation of Network B after feature extraction by the encoder. Since J and R are scalar quantities rather than spatially varying fields, introducing them at the latent level enables the convolutional encoder to first extract spatial features from the difference flow fields before conditioning the latent representation on the governing flow parameters.

The normalized governing flow parameters are given by

$$J^*_{(i)} = \frac{J_{(i)}}{J_{\max}}, \qquad R^*_{(i)} = \frac{R_{(i)}}{R_{\max}}, \tag{28}$$

where $J_{(i)}$ and $R_{(i)}$ denote the centerline gradient Richardson number and scale ratio, respectively, corresponding to the ith flow snapshot. The quantities $J_{\max}$ and $R_{\max}$ denote the maximum values of the corresponding parameters in the dataset. The latent representation obtained from Eq. (27) is subsequently flattened into a one-dimensional latent vector,

$$\mathbf{d}^f_{(i)} \in \mathbb{R}^{1024}, \tag{29}$$

where $\mathbf{d}^f_{(i)}$ denotes the flattened one-dimensional latent vector corresponding to the encoded difference flow fields. The normalized governing flow parameters, $J^*_{(i)}$ and $R^*_{(i)}$, are concatenated with the latent vector to form the augmented latent vector,

$$\mathbf{d}^a_{(i)} = \left[\mathbf{d}^f_{(i)}, J^*_{(i)}, R^*_{(i)}\right], \qquad \mathbf{d}^a_{(i)} \in \mathbb{R}^{1026}, \tag{30}$$

where $\mathbf{d}^a_{(i)}$ denotes the parameter-augmented latent vector. The augmented latent vector ($\mathbf{d}^a_{(i)}$) is mapped back to the original latent-space dimension through a FCL, yielding

$$\widetilde{\mathbf{d}}^f_{(i)} = \mathbf{W}_{\mathrm{fc}}\mathbf{d}^a_{(i)} + \mathbf{b}_{\mathrm{fc}}, \qquad \widetilde{\mathbf{d}}^f_{(i)} \in \mathbb{R}^{1024}, \tag{31}$$

where $\widetilde{\mathbf{d}}^f_{(i)}$ denotes the parameter-conditioned latent vector after the FCL, and $\mathbf{W}_{\mathrm{fc}}$ and $\mathbf{b}_{\mathrm{fc}}$ denote the learnable weight matrix and bias vector, respectively. The parameter-conditioned latent vector is subsequently restored to the original latent-space dimensions to obtain the final parameter-conditioned latent representation, denoted as $\mathbf{d}(\mathbf{lv})_{(i)}$,

$$\mathbf{d}(\mathbf{lv})_{(i)} \in \mathbb{R}^{4\times8\times32}, \tag{32}$$

The decoder, $\mathscr{D}$, reconstructs the normalized difference flow fields from the final parameter-conditioned latent representation,

$$\Delta\hat{\mathbf{X}}^*_{(i)} = \mathscr{D}\left(\mathbf{d}(\mathbf{lv})_{(i)}\right), \qquad \Delta\hat{\mathbf{X}}^*_{(i)} = \left(\Delta\hat{\Omega}^*_{(i)}, \Delta\hat{B}^*_{(i)}\right) \in \mathbb{R}^{128\times256\times2}, \tag{33}$$

where $\Delta\hat{\mathbf{X}}^*_{(i)}$ denotes the reconstructed normalized difference flow snapshot, whose two channels correspond to the reconstructed normalized difference vorticity, $\Delta\hat{\Omega}^*_{(i)}$, and reconstructed normalized difference buoyancy, $\Delta\hat{B}^*_{(i)}$, respectively. The reconstructed normalized difference flow fields are subsequently rescaled to obtain the corresponding physical-space difference flow fields,

$$\Delta\hat{\mathbf{X}}_{(i)} = \left(\Delta\hat{\Omega}_{(i)}, \Delta\hat{B}_{(i)}\right) \in \mathbb{R}^{128\times256\times2}, \tag{34}$$

where $\Delta\hat{\mathbf{X}}_{(i)}$ denotes the reconstructed physical-space difference flow snapshot.

Network B is trained using the loss function defined in Eq. (15), applied to the normalized difference flow fields. By learning the difference flow fields rather than the absolute flow fields, Network B focuses on representing the small-scale, transient, and intermittent flow motions that are not captured by Network A.

3. *LSTM Model : Network C*

Network C is designed to learn the temporal evolution of KH instabilities across different flow conditions characterized by J and R. Unlike the traditional LSTM model, which operates on the latent representation obtained from a single CNN-AE, the proposed Network C utilizes the scale-separated latent representations extracted from Networks A and B. Once Networks A and B are trained, the state latent representation, $\mathbf{lv}_{(i)}$,

learned by Network A and the parameter-conditioned difference latent representation, $\mathbf{d}(\mathbf{lv})_{(i)}$, learned by Network B are concatenated. The resulting latent representation is subsequently transformed into a one-dimensional latent vector, which serves as the input to Network C.

Since the latent representations from Networks A and B encode different scales of the flow, they exhibit different value ranges. Direct concatenation can therefore lead to an imbalanced input representation, where high-amplitude small-scale features may dominate the temporal learning process. To ensure balanced contributions from both latent components, each latent representation is independently normalized using mean-standard deviation normalization prior to concatenation.

A mean-standard deviation normalization is applied independently to the latent representations obtained from Networks A and B before training the LSTM model. The normalized latent representations are given by

$$\widetilde{\mathbf{lv}}_{(i)} = \frac{\mathbf{lv}_{(i)} - \boldsymbol{\mu}_{lv}}{\boldsymbol{\sigma}_{lv}}, \qquad \widetilde{\mathbf{d}(\mathbf{lv})}_{(i)} = \frac{\mathbf{d}(\mathbf{lv})_{(i)} - \boldsymbol{\mu}_{d}}{\boldsymbol{\sigma}_{d}}, \tag{35}$$

where $\widetilde{\mathbf{lv}}_{(i)}$ and $\widetilde{\mathbf{d}(\mathbf{lv})}_{(i)}$ denote the normalized state and difference latent representations of Networks A and B, respectively.

The mean and standard deviation are computed element-wise across all latent representations in the training dataset. Specifically, for each spatial location and channel in the latent representation, the corresponding values from all training samples, aggregated over all training cases and time snapshots, are used to compute a single mean and standard deviation. The statistics for the latent representations of Network A are computed as

$$\boldsymbol{\mu}_{lv} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{lv}^{(n)}, \qquad \boldsymbol{\sigma}_{lv} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(\mathbf{lv}^{(n)} - \boldsymbol{\mu}_{lv}\right)^2}, \tag{36}$$

while the corresponding statistics for the latent representations of Network B are computed as

$$\boldsymbol{\mu}_{d} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{d}(\mathbf{lv})^{(n)}, \quad \boldsymbol{\sigma}_{d} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(\mathbf{d}(\mathbf{lv})^{(n)} - \boldsymbol{\mu}_{d}\right)^2}, \tag{37}$$

where $\boldsymbol{\mu}_{lv}$ and $\boldsymbol{\sigma}_{lv}$ denote the element-wise mean and standard deviation of the latent representations from Network A; $\boldsymbol{\mu}_{d}$ and $\boldsymbol{\sigma}_{d}$ denote the corresponding element-wise mean and standard deviation of the latent representations from Network B. Furthermore, $\mathbf{lv}^{(n)} \in \mathbb{R}^{4\times8\times16}$ and $\mathbf{d}(\mathbf{lv})^{(n)} \in \mathbb{R}^{4\times8\times32}$ denote the latent representations corresponding to the nth training sample from Networks A and B, respectively, N is the total number of latent representations aggregated over all training cases and time snapshots. Consequently, the mean and standard deviation tensors have the same dimensions as their corresponding latent representations, i.e., $\boldsymbol{\mu}_{lv}, \boldsymbol{\sigma}_{lv} \in \mathbb{R}^{4\times8\times16}$ and $\boldsymbol{\mu}_{d}, \boldsymbol{\sigma}_{d} \in \mathbb{R}^{4\times8\times32}$.

The vectors $\widetilde{\mathbf{lv}}_{(i)}$ and $\widetilde{\mathbf{d}(\mathbf{lv})}_{(i)}$ are flattened and concatenated into a one-dimensional latent vector, denoted by

$$\mathbf{Z}_{(i)} \in \mathbb{R}^{1536}, \tag{38}$$

which serves as the input to Network C.

Network C employs the same parameter-conditioned LSTM architecture described in Sec. III A 2. The latent vector, $\mathbf{Z}_{(i)}$, is augmented with the governing flow parameters, $J_{(i)}$ and $R_{(i)}$, to construct the input vector, $\mathbf{h}_{(i)}$, and is subsequently propagated through the LSTM using the same one-step prediction strategy and recursive inference procedure described in Eqs. (19)–(23). The primary architectural difference from the traditional LSTM framework is that the latent-space dimension increases from 512 to 1536 owing to the concatenation of the state and parameter-conditioned difference latent representations.

Network C employs the same training configuration as the traditional LSTM model, except for the input, hidden, and output dimensions and the learning rate. The input, hidden, and output dimensions are increased to 1538, 1536, and 1536, respectively, while the learning rate is reduced to 10^{-4}. The model is trained for 3000 epochs, consistent with the traditional LSTM model. The optimized hyperparameters are summarized in Table IV.

Although the LSTM output and latent-space dimensions are both 1536, a fully connected output layer is employed to provide additional flexibility in mapping the LSTM output to the latent representation required by the decoders. To assess its contribution, an ablation study was performed by training Network C with and without the fully connected output layer while keeping all other architectural components and training settings unchanged. The comparison presented in Appendix C shows that removing the fully connected output layer increases the prediction error, particularly during the nonlinear evolution of the KH instability. Therefore, the fully connected output layer is retained in the final architecture.

The predicted latent vector is subsequently reshaped to its original dimensions and separated into the state latent representation and the parameter-conditioned difference latent representation. These latent representations are supplied to the corresponding decoders of Networks A and B to reconstruct the large-scale and small-scale flow components, respectively. Finally, the reconstructed small-scale flow fields are superimposed onto the reconstructed large-scale flow fields to obtain the predicted flow snapshot. This reconstruction strategy preserves both dominant coherent structures and fine-scale flow features, thereby enabling physically consistent prediction of multiscale flow dynamics. The overall framework integrates spatial encoding, scale separation, temporal prediction, and reconstruction within a unified deep-learning architecture.

IV. RESULTS

In this section, we assess the performance of the traditional CNN-AE + LSTM model and the proposed multiscale CNN-AE + LSTM model in predicting the temporal evolution of KH instability from a given initial snapshot. The comparison is conducted over various flow conditions defined in the parameter space of J and R, encompassing both training and unseen test cases. For detailed evaluation, three representative training cases are selected from the 32 training cases used during

model development, along with the three unseen testing cases introduced in Section II C. The selected representative cases are summarized in Table V.

TABLE V. Representative training and testing cases selected for detailed model evaluation.

Case	R	J	Description
Training 1	3.0	0.05	Model seen R and J
Training 2	5.0	0.15	Model seen R and J
Training 3	7.0	0.20	Model seen R and J
Testing 1	2.5	0.05	Model unseen R, seen J
Testing 2	7.0	0.18	Model seen R, unseen J
Testing 3	4.5	0.12	Model unseen R and J

The models are assessed based on their ability to accurately reconstruct flow fields and capture the underlying dynamics. In addition to snapshot comparison, physics-based metrics, such as perturbation enstrophy, growth rate, and the 2D kinetic energy spectrum, are used to quantify predictive performance. The results are presented in the following order: first, the performance of the traditional model is analyzed to highlight its limitations, followed by the evaluation of the proposed scale-separated framework and its improvements over the traditional approach.

A. Traditional CNN-AE + LSTM Model

Before training, the input flow fields are scaled so as to ensure that the values of both Ω and B are in the range $[-1,1]$. The scaled flow fields are then passed through the CNN-AE model, where they are compressed from an input size of $128\times256\times2$ to a latent representation of size $4\times8\times16$, reducing the dimensionality by a factor of 128 while retaining the dominant flow features. These latent representations are subsequently provided to the LSTM model to learn and predict the temporal evolution of the flow. The predicted latent vectors are then decoded and rescaled to reconstruct the physical flow fields. The performance of the traditional CNN-AE + LSTM framework is evaluated based on its ability to reconstruct flow fields and predict their temporal evolution across different flow conditions characterized by the parameters J and R.

The training and validation loss histories of the traditional framework are presented in Fig. 6, while the final training and validation losses of both the traditional and proposed multi-scale frameworks are summarized in Table VI. The loss histories demonstrate stable convergence for both the CNN-AE and LSTM models, characterized by a rapid decrease during the initial stages of training followed by a gradual plateau. As shown in Fig. 6(a), the CNN-AE converges after approximately 8000 training epochs, whereas the LSTM model reaches convergence within approximately 3000 epochs, as shown in Fig. 6(b). The final training and validation losses indicate satisfactory model performance, with the LSTM exhibiting lower loss values owing to learning in the reduced-order latent space. Beyond these training epochs, further optimization resulted in only marginal improvements in the validation loss, indicating that both models had reached convergence.

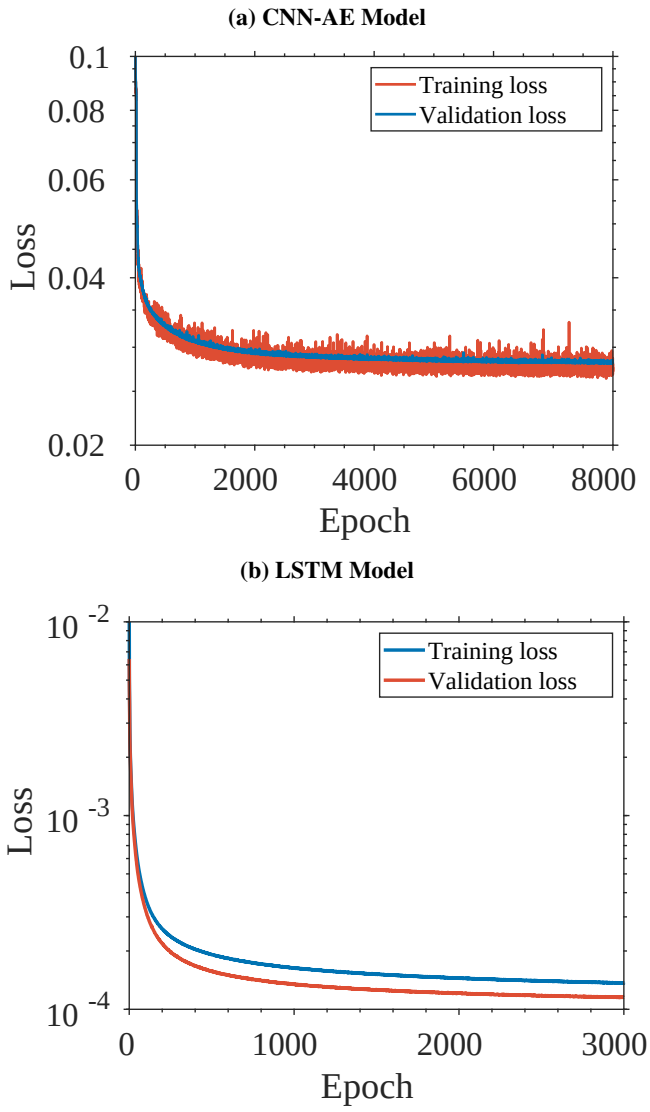


FIG. 6. Training and validation loss histories of the traditional CNN-AE + LSTM framework. (a) CNN-AE model trained for 8000 epochs. (b) LSTM model trained for 3000 epochs using the latent representations extracted from the CNN-AE. Both plots are shown on a logarithmic scale to illustrate the convergence behavior during training.

The trained CNN-AE + LSTM framework is evaluated using representative training and unseen testing cases to assess its predictive capability across different flow conditions. Starting from an initial flow field at approximately two e-folding times ($t = 20$ s), the model recursively predicts the evolution of Ω and B up to $t = 200$ s. Fig. 7 compares the LES flow fields with the corresponding CNN-AE + LSTM predictions for a representative training case 2 and an unseen testing case 3. At early times, the model accurately reproduces the initial shear-layer deformation and vortex roll-up structures with rel-

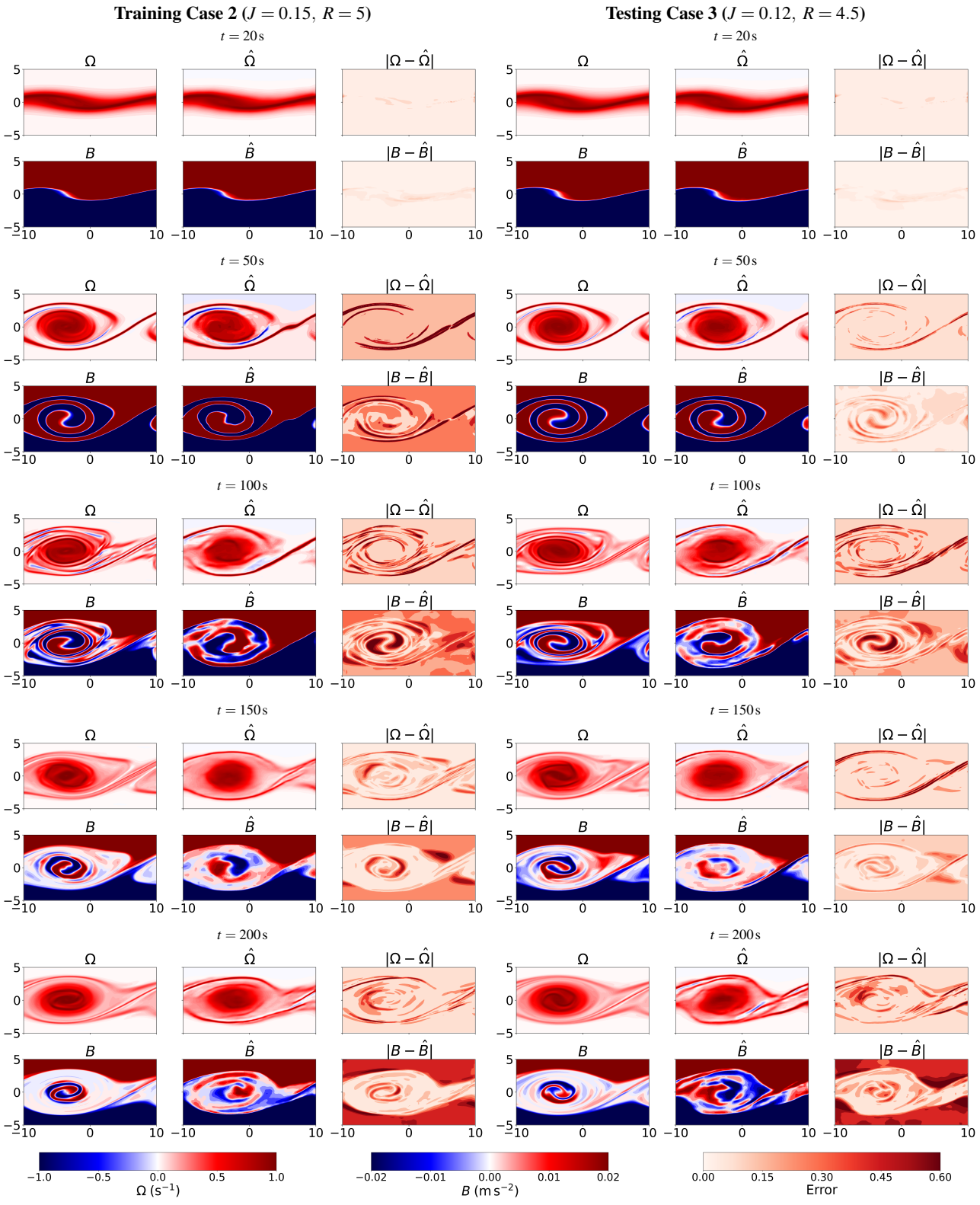


FIG. 7. Comparison of LES and traditional CNN-AE + LSTM predictions for a representative case. The left column corresponds to training case 2, while the right column shows predictions for unseen testing case 3 at $t = 20$, 50, 100, 150, and 200 s. Within each panel, the first, second, and third sub-images represent the LES field, reconstructed field, and absolute error, respectively. The top and bottom sub-images correspond to vorticity and buoyancy fields. Here, Ω and B denote the LES vorticity and buoyancy fields, $\hat{\Omega}$ and $\hat{B}$ denote the reconstructed fields, and $|\Omega - \hat{\Omega}|$ and $|B - \hat{B}|$ represent the corresponding absolute errors. Although the model captures the large-scale flow evolution reasonably well, the reconstructed fields become progressively smoother at later times, leading to loss of fine-scale vortical structures and sharp buoyancy gradients during the nonlinear stage.

atively small reconstruction errors. During the intermediate nonlinear stage ($t = 50$), the framework successfully captures the dominant coherent structures and overall spatial organization of the flow despite the substantial dimensionality reduction introduced by the latent-space representation.

As the flow evolves into the strongly nonlinear regime ($t =$ 150 and 200 s), the reconstructed fields become progressively smoother, particularly in regions containing fine-scale vortical structures and sharp buoyancy gradients. Consequently, the smaller-scale turbulent features weaken relative to the LES data, resulting in increased reconstruction errors at later times. Similar behavior is observed for the unseen testing case, indicating that the model generalizes reasonably well across different combinations of J and R.

The traditional CNN-AE + LSTM framework is able to capture the main, large-scale temporal evolution of KH instability and shows good generalization across different flow conditions. With only a single flow snapshot and the governing flow parameters as input, it can recursively predict the flow evolution over long time horizons. Yet, as the flow enters the strongly nonlinear stage, fine-scale vortical structures and sharp buoyancy gradients are gradually lost in the reconstructed fields. This loss of small-scale detail reveals the limits of using a single-scale latent representation for spatial reconstruction and, in turn, motivates the multiscale framework introduced in this work.

B. Proposed Multi-Scale CNN-AE and LSTM Model

The proposed framework is evaluated using the same training and testing cases as those considered for the traditional CNN-AE + LSTM model, thereby facilitating a direct assessment of the benefits of scale-separated feature learning. The preprocessing, scaling, and training configuration of the proposed networks are described in Section III B.

Since Network A adopts the same architecture and training configuration as the traditional CNN-AE model, its training and validation loss histories are identical to those of the traditional CNN-AE model and are presented in Fig. 6(a). The corresponding loss histories for Network B (Difference CNN-AE) and Network C (LSTM) are shown in Fig. 8(a) and (b), respectively, while the converged training and validation losses of all three networks are summarized in Table VI.

TABLE VI. Summary of the converged training and validation losses for the traditional and proposed models. The reported loss values correspond to the converged loss levels obtained at the end of training. Network A is identical to the CNN-AE model used in the traditional framework and is therefore reported together with the CNN-AE results.

Metric	CNN-AE (Network A)	LSTM Model	Network B	Network C
Convergence epoch	8000	3000	15000	3000
Training loss	2.8×10^{-2}	1.5×10^{-4}	3.8×10^{-4}	3.5×10^{-2}
Validation loss	2.8×10^{-2}	1.1×10^{-4}	5.9×10^{-4}	1.0×10^{-1}

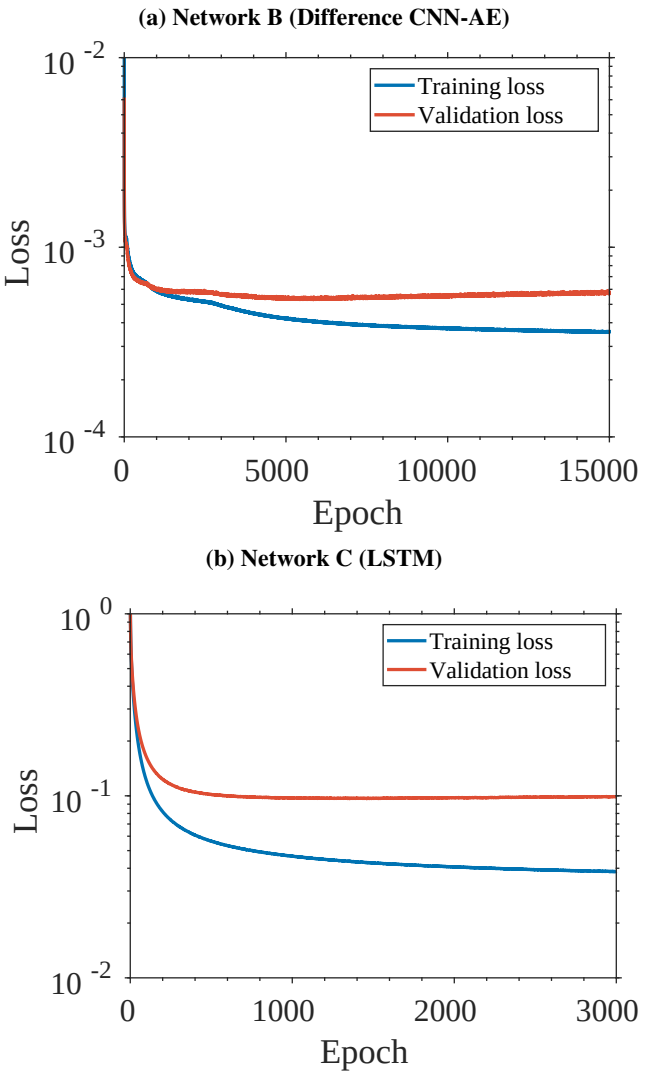


FIG. 8. Training and validation loss histories of the proposed multiscale framework. (a) Difference CNN-AE (Network B) trained for 15,000 epochs to reconstruct the difference flow fields. (b) LSTM (Network C) trained for 3,000 epochs to predict the temporal evolution of the combined latent representations. Both plots are presented on a logarithmic scale to illustrate the convergence behavior during training.

Network B exhibits stable convergence, with training and validation losses remaining relatively close throughout the training process, indicating good reconstruction capability and generalization. Network C exhibits a noticeable gap between the training and validation losses throughout the training process. The training loss continues to decrease with increasing epochs, whereas the validation loss decreases initially and subsequently reaches a near-constant value without exhibiting an upward trend. This behavior indicates that the model continues to improve its fit to the training data while maintaining stable validation performance, suggesting that no significant overfitting is observed within the considered training range.

Compared with the traditional CNN-AE + LSTM frame-

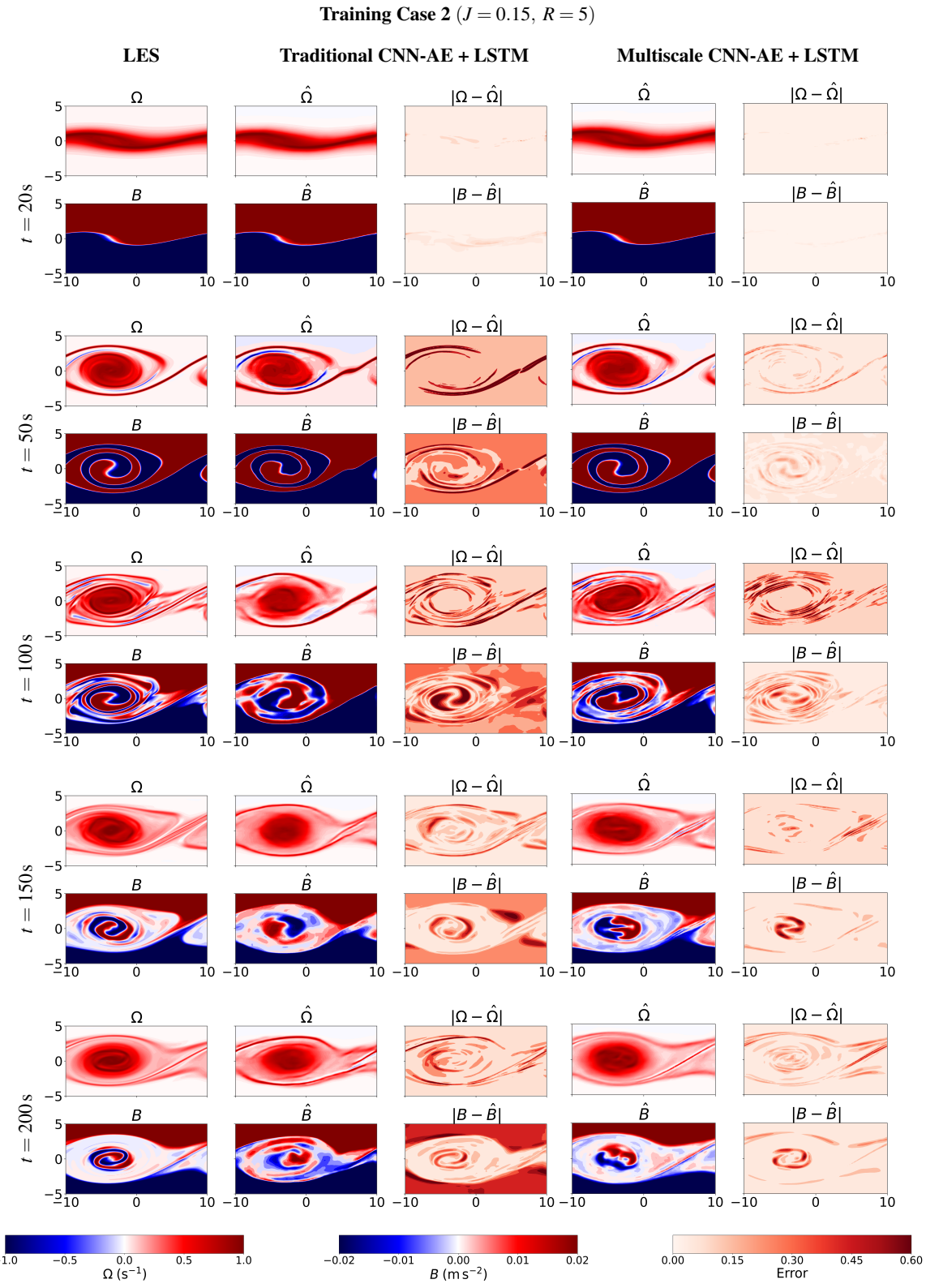


FIG. 9. Comparison between the traditional CNN-AE + LSTM framework and the proposed multiscale CNN-AE + LSTM framework for Training Case 2 ($J = 0.15$, $R = 5$) at different time instants. The left column shows the LES solution together with the traditional CNN-AE + LSTM reconstruction and corresponding absolute error, while the right column shows the proposed multiscale framework reconstruction and error. The rows correspond to different time instants from $t = 20$ s to $t = 200$ s. Within each sub-image, the top and bottom panels represent vorticity and buoyancy fields, respectively. Here, Ω and B denote the LES vorticity and buoyancy fields, $\hat{\Omega}$ and $\hat{B}$ denote the reconstructed fields, and $|\Omega - \hat{\Omega}|$ and $|B - \hat{B}|$ represent the corresponding absolute errors.

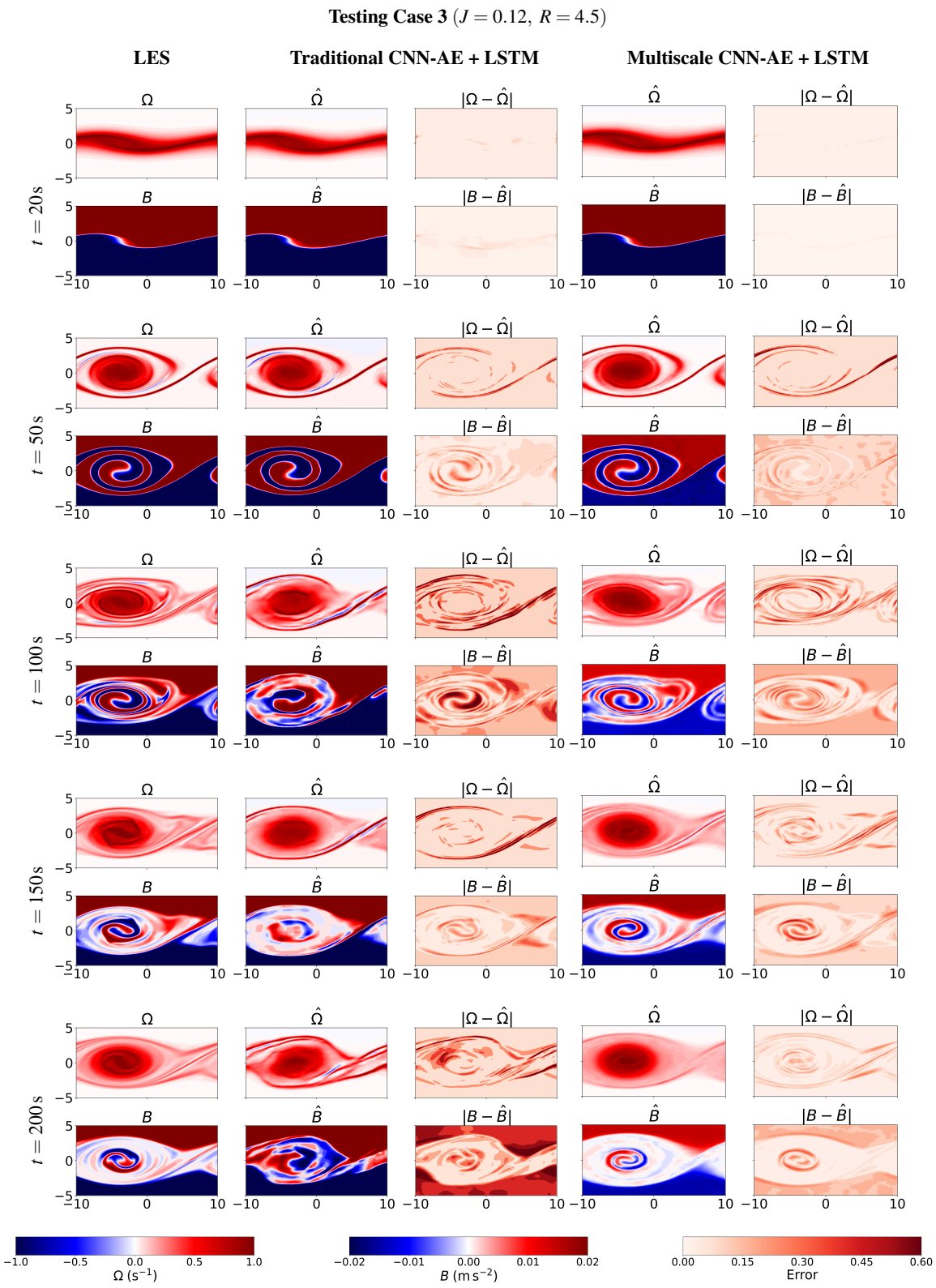


FIG. 10. Same as Fig. 9, but for Testing case 3.

work (Figs. 7), the proposed multiscale framework (Fig. 9 and 10) shows improved reconstruction of the evolving KH instability for both the training and unseen testing cases. Starting from an initial snapshot ($t = 20$ s) for specified (J,R) conditions, both frameworks accurately capture the initial shear-layer structure and vortex roll-up at early times. However, noticeable differences emerge as the flow evolves into the nonlinear stage. At intermediate times ($t = 50$ s and $t = 100$ s), the traditional CNN-AE + LSTM framework begins to produce smoother vorticity and buoyancy fields, resulting in partial loss of fine-scale spiral structures within the vortex core and reduced sharpness along the shear interface. The associated reconstruction error also becomes more spatially distributed, particularly near regions with strong gradients. These differences become more pronounced at later times ($t = 150$ s and $t = 200$ s), where the traditional framework exhibits significant smoothing of vortical structures and buoyancy gradients. In contrast, the proposed multiscale framework preserves sharper interface structures and multiscale vortical features more effectively, while maintaining lower and more localized reconstruction error throughout the flow evolution.

With the flow parameters (J,R) used as conditioning inputs, the model learns flow evolution patterns across instability regimes and can predict the full temporal evolution of KH instability from a single initial snapshot. These findings indicate that the scale-separated framework generalizes across flow conditions and captures complex multiscale dynamics directly from data, without numerically solving the governing equations.

C. Performance Assessment of CNN-AE + LSTM Models

This section evaluates the predictive performance of the traditional CNN-AE + LSTM framework and the proposed multiscale CNN-AE + LSTM framework using both statistical error metrics and physics-based flow diagnostics.

1. *Performance Metrics*

The predictive performance of the proposed framework is evaluated using the mean squared error (MSE) and the coefficient of determination (R^2). These metrics are computed for each flow snapshot by comparing the LES reference and the model-predicted flow fields. For each snapshot, the reconstructed vorticity and buoyancy fields are combined and flattened into a single vector, and the error metrics are subsequently evaluated over all spatial grid points and both flow variables. The temporal evolution of the snapshot-wise MSE is further examined for representative training and testing cases to assess the variation of the prediction error throughout the evolution of the flow. The MSE and R^2 are defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{39}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{40}$$

where y_i and $\hat{y}_i$ denote the LES reference and model-predicted values, respectively, $\bar{y}$ is the mean of the LES reference values, and N is the total number of values obtained by combining the vorticity and buoyancy fields over all spatial grid points in a given flow snapshot.

TABLE VII. Comparison of the overall prediction performance of the traditional CNN-AE + LSTM and the proposed multiscale CNN-AE + LSTM frameworks for the representative cases listed in Table V. The reported MSE and R^2 values are averaged over all temporal snapshots and evaluated over the combined vorticity (Ω) and buoyancy (B) flow fields. Lower MSE values and higher R^2 values indicate better predictive performance.

Case	MSE		R^2	
	Traditional	Proposed	Traditional	Proposed
Training Case 1	1.456×10^{-2}	1.233×10^{-3}	0.668	0.973
Training Case 2	2.072×10^{-2}	7.131×10^{-3}	0.603	0.860
Training Case 3	6.996×10^{-3}	3.465×10^{-3}	0.859	0.929
Testing Case 1	8.730×10^{-3}	1.750×10^{-3}	0.810	0.962
Testing Case 2	1.303×10^{-2}	6.347×10^{-3}	0.736	0.871
Testing Case 3	5.607×10^{-3}	3.916×10^{-3}	0.883	0.918

Fig. 11(a)–(c) presents the temporal evolution of the MSE computed over the combined Ω and B fields for the representative training cases, while Fig. 11(d)–(f) presents the corresponding results for the representative testing cases. The proposed multiscale framework consistently achieves lower and more stable MSE values than the traditional approach. The differences become particularly pronounced during the nonlinear stages of instability development, where the traditional model exhibits substantial error. Correspondingly, the proposed framework yields higher R^2 values across all cases, with the largest improvements observed for Training Cases 1 and 2 [Figs. 11(a) and 11(b)] and Testing Cases 1 and 2 [Figs. 11(d) and 11(e)], which are characterized by stronger nonlinear and multiscale dynamics (Table VII).

The mean R^2 increases from 0.760 for the traditional framework to 0.919 for the proposed multiscale framework (Table VII), indicating a substantial improvement in predictive accuracy. These results demonstrate the ability of the proposed framework to more accurately predict the evolution of complex stratified shear flow instabilities.

2. *Physics-Based Performance Assessment*

While statistical metrics such as MSE and R^2 provide quantitative measures of reconstruction accuracy, they do not necessarily assess whether the predicted flow fields preserve the underlying physics of the instability. The predictive capability

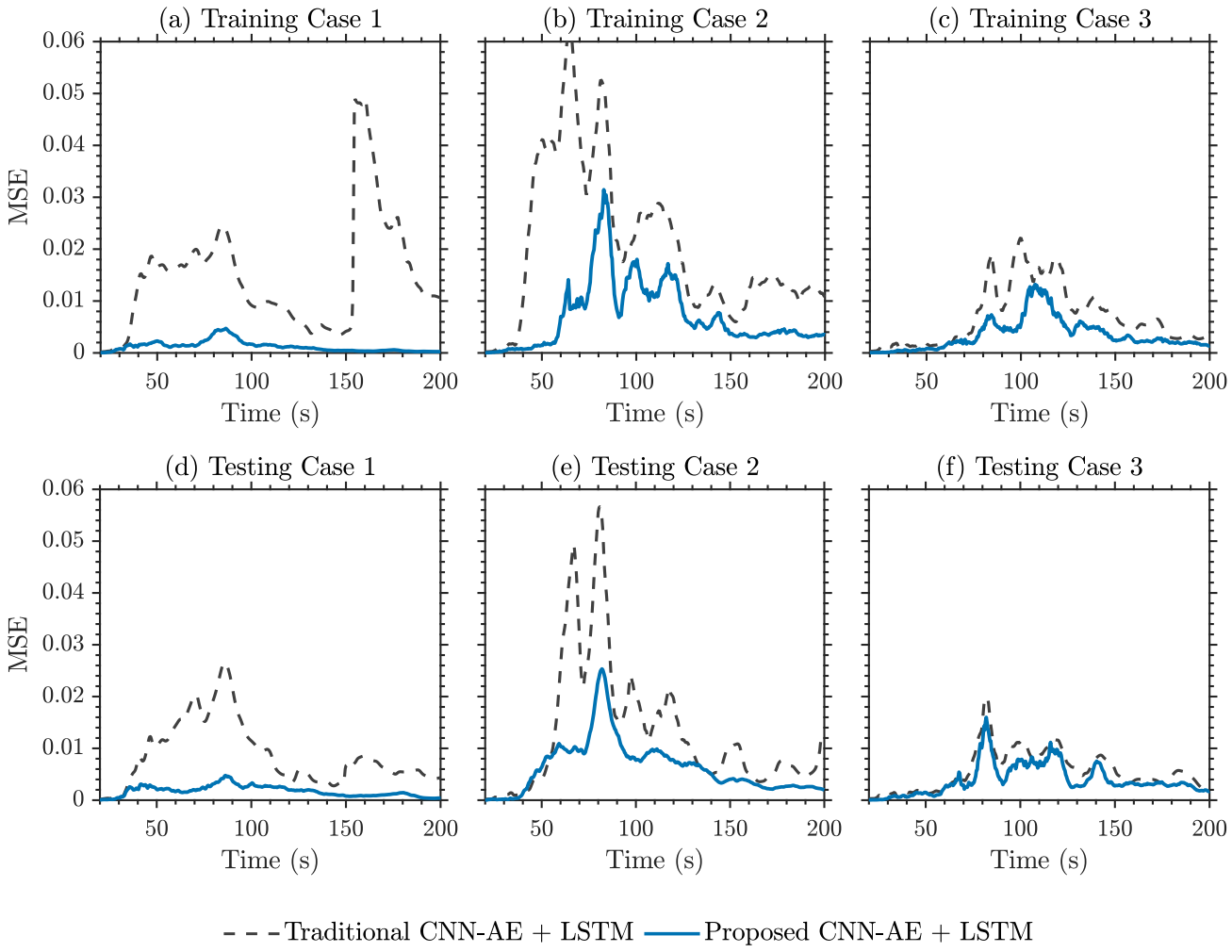


FIG. 11. Temporal evolution of the MSE for the representative training and testing cases listed in Table V. Note that this is the total MSE computed for the Ω and B fields. The upper panel corresponds to the training cases, while the lower panel corresponds to the testing cases. The dashed line denotes the traditional CNN-AE + LSTM framework, and the solid line represents the proposed multiscale CNN-AE + LSTM framework.

of both frameworks is further evaluated using physics-based flow diagnostics, including perturbation enstrophy, instability growth rate, and kinetic energy spectra, and compared directly with the corresponding LES results.

a. ***Perturbation Enstrophy:*** The perturbation enstrophy provides a quantitative measure of the strength and evolution of vortical motions within the flow. It is defined as

$$\Omega_{\text{pert}}(t) = \frac{1}{2}\int_A \Omega'^2\, dx\, dz, \tag{41}$$

, where $\Omega_{\text{pert}}(t)$ denotes the perturbation enstrophy at time t, Ω' is the perturbation vorticity obtained by subtracting the base-flow vorticity from the instantaneous vorticity field, and A denotes the 2D computational domain.

Fig. 12(a)–(f) compares the evolution of perturbation enstrophy predicted by the traditional CNN-AE + LSTM and proposed multiscale CNN-AE + LSTM models against the LES results for the selected training and testing cases. Both approaches capture the overall growth and subsequent nonlinear evolution of the instability. During the early stages, the predicted enstrophy closely follows the LES evolution, indicating that both models adequately represent the initial amplification of perturbations.

As the flow transitions into the nonlinear regime, differences between the two approaches become more pronounced. The multiscale model maintains closer agreement with the LES across all cases, reproducing both the magnitude and temporal evolution of the enstrophy with greater accuracy. In contrast, the traditional model exhibits larger deviations and phase mismatches.

The improved performance of the multiscale model can be attributed to its scale-separated representation, which facilitates the preservation of fine-scale vortical structures throughout the prediction horizon. As a result, it more accurately represents the physical processes associated with KH instability. Overall, these results indicate that the proposed frame-

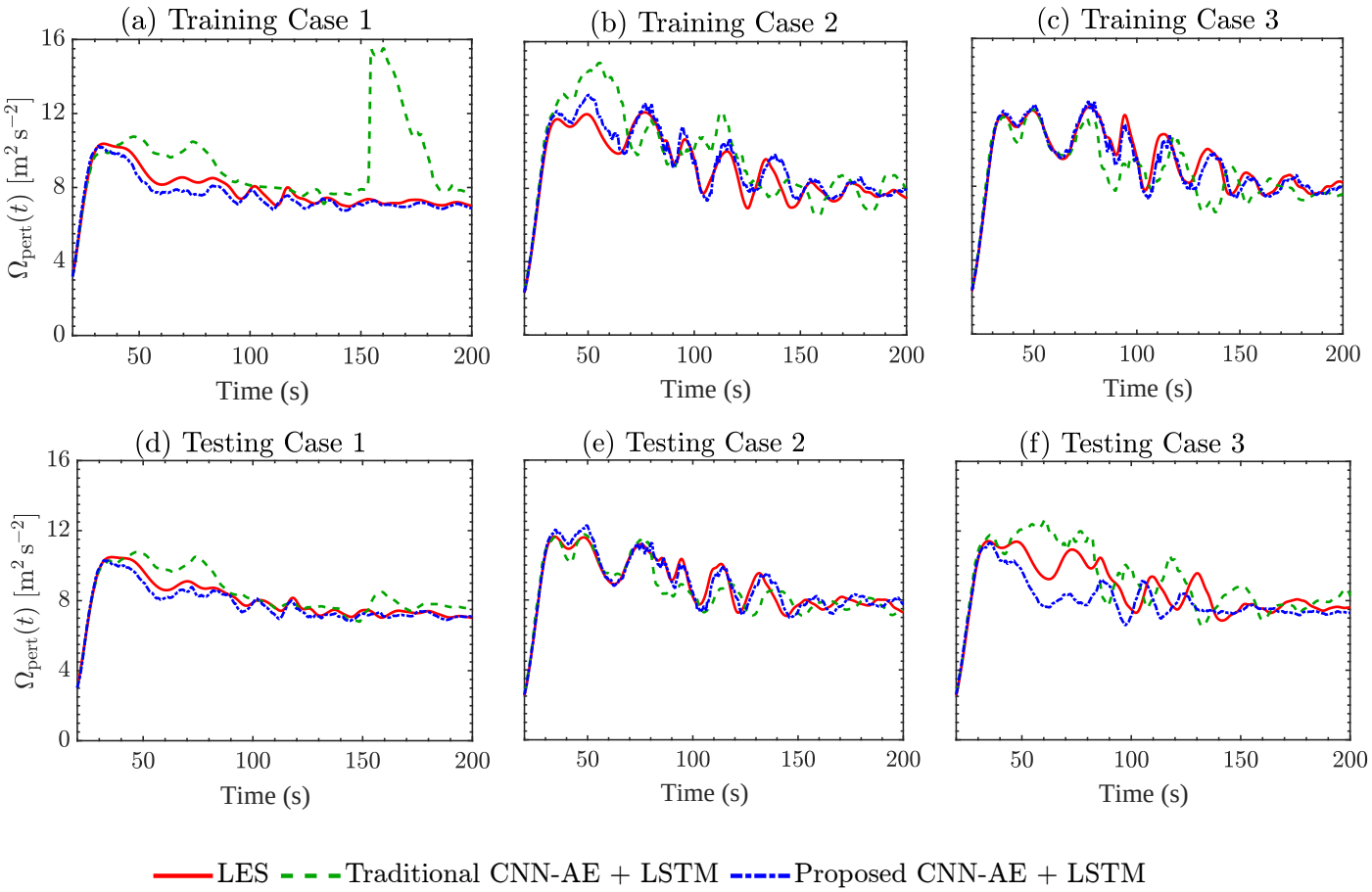


FIG. 12. Comparison of area-integrated perturbation enstrophy evolution for the representative training and testing cases (Table V) using LES, the traditional CNN-AE + LSTM framework, and the proposed multiscale CNN-AE + LSTM framework.

work provides a more accurate representation of the instability dynamics across a range of flow conditions characterized by different values of J and R.

b. ***Growth Rate:*** The instability growth rate is inferred from the exponential growth of perturbation enstrophy. During the linear regime,

$$\Omega_{\text{pert}}(t) \sim e^{2\sigma t}, \tag{42}$$

where σ denotes the instability growth rate. It is obtained from the slope of $\log(\Omega_{\text{pert}}(t))$ within the linear growth regime. Since the growth rate characterizes the amplification of perturbations during the early stages of KH instability, it provides a useful measure for assessing whether the ROM correctly reproduces the underlying linear instability dynamics.

Table VIII compares the instability growth rates obtained from LES and the ROMs. Both the traditional CNN-AE + LSTM framework and the proposed multiscale CNN-AE + LSTM framework accurately reproduce the LES growth rates for the selected training and testing cases, demonstrating that the dominant linear instability dynamics are well captured. Because the growth rate is evaluated within the linear regime, differences between the two frameworks are relatively small. The advantages of the proposed multiscale framework become more apparent during the nonlinear evolution of the flow, as discussed in the following sections.

TABLE VIII. Comparison of instability growth rates predicted using LES, the traditional CNN-AE + LSTM framework, and the proposed multiscale CNN-AE + LSTM framework for the representative cases.

Case	LES	Traditional	Multiscale
Training Case 1	0.101	0.099	0.108
Training Case 2	0.109	0.102	0.107
Training Case 3	0.109	0.101	0.111
Testing Case 1	0.101	0.099	0.080
Testing Case 2	0.109	0.108	0.100
Testing Case 3	0.109	0.105	0.106

c. ***2D Kinetic Energy Spectrum:*** The kinetic energy spectrum provides a quantitative measure of how kinetic energy is distributed across spatial scales and therefore serves as an effective diagnostic for evaluating the ability of reduced-order models to preserve both large-scale and fine-scale flow structures.

Since the instability dynamics are contained in the perturbation field, the kinetic energy spectrum is computed using the perturbation velocity components. The 2D discrete

Fourier transform of the perturbation velocity field $\mathbf{u}(\mathbf{x},t) = [u(x,z,t), w(x,z,t)]$ is defined as

$$\hat{\mathbf{u}}(\mathbf{k},t) = \int_{\mathscr{D}} \mathbf{u}(\mathbf{x},t) e^{-i\mathbf{k}\cdot\mathbf{x}} d\mathbf{x}, \tag{43}$$

where $i = \sqrt{-1}$, $\mathbf{x} = (x,z)$ denotes the spatial coordinate vector, $\mathbf{k} = (k_x, k_z)$ is the 2D wavevector, and $\mathscr{D}$ represents the computational domain. Prior to computing the Fourier transform, the spatial mean of each velocity component is removed and a Hann window is applied in both spatial directions to reduce spectral leakage.

The corresponding 2D kinetic energy density is given by

$$E_{2D}(\mathbf{k},t) = \frac{1}{2}\left(|\hat{u}(\mathbf{k},t)|^2 + |\hat{w}(\mathbf{k},t)|^2\right), \tag{44}$$

where $\hat{u}$ and $\hat{w}$ denote the Fourier coefficients of the streamwise and vertical perturbation velocity components, respectively.

The 2D spectra obtained from individual snapshots are first averaged over all snapshots within the selected time interval,

$$\overline{E}_{2D}(\mathbf{k}) = \frac{1}{N_t} \sum_{m=1}^{N_t} E_{2D}^{(m)}(\mathbf{k}), \tag{45}$$

where N_t is the total number of snapshots within the averaging interval and $E_{2D}^{(m)}$ denotes the 2D kinetic energy density corresponding to the mth snapshot.

A radial integration of the time-averaged 2D kinetic energy spectrum,

$$E(K_n) = \sum_{K_n \le \sqrt{k_x^2+k_z^2} < K_n + \Delta K} \overline{E}_{2D}(\mathbf{k}), \tag{46}$$

where $K = \sqrt{k_x^2 + k_z^2}$ is the radial wavenumber, K_n denotes the radial wavenumber of the nth shell, and ΔK is the shell width used for spectral binning. The 2D kinetic energy spectrum is obtained by integrating the time-averaged spectral energy over concentric wavenumber shells in Fourier space.[39]

Fig. 13 compares the time-averaged kinetic energy spectra obtained from LES, the traditional CNN-AE + LSTM framework, and the proposed multiscale CNN-AE + LSTM framework for the selected training and testing cases. The spectra are shown for two representative stages of KH evolution: the early nonlinear stage (80–100 s; Fig. 13(a), (c), (e), (g), (i), and (k)) and the later nonlinear stage (180–200 s; Fig. 13(b), (d), (f), (h), (j), and (l)).

During the early nonlinear stage, both reduced-order frameworks accurately reproduce the LES spectra over the energy-containing range ($k \lesssim 4\,\mathrm{rad/m}$). At this stage, the flow dynamics are dominated by the primary KH billows, and the majority of the kinetic energy remains concentrated at large scales. Consequently, the spectra predicted by both frameworks closely follow the LES results. Minor deviations emerge at intermediate wavenumbers, where the traditional CNN-AE + LSTM framework exhibits a slightly faster decay, whereas the multiscale framework remains in closer agreement with the LES spectra.

More substantial differences appear during the later nonlinear stage. As the KH billows undergo roll-up and filamentation, energy is redistributed over a broader range of spatial scales and increasingly fine-scale vortical structures are generated. While both frameworks continue to reproduce the large-scale motions, the traditional CNN-AE + LSTM framework exhibits noticeable distortions in the spectral distribution during the nonlinear stage, particularly for the testing cases shown in Fig. 13(d), (h) and (l). In these cases, excess energy appears at intermediate wavenumbers, producing a spectral shape that differs significantly from the LES results. In contrast, the proposed multiscale CNN-AE + LSTM framework remains in close agreement with the LES spectra across the entire resolved wavenumber range and more accurately reproduces the spectral decay. These results indicate that the multiscale decomposition improves the representation of scale interactions occurring during the nonlinear evolution of the instability.

The LES spectra exhibit an intermediate-range decay close to a $k^{-5/3}$ scaling during the early nonlinear stage and evolve toward a steeper k^{-3} scaling during the later stage. Similar multiscale dynamics have been reported in DNS studies of KH instability[40]. Consequently, the spectrum steepens and approaches a k^{-3} scaling during the later stage. In 2D turbulence, such behavior is commonly associated with the increasing influence of forward enstrophy transfer toward smaller scales[41]. The observed transition from a spectrum approaching $k^{-5/3}$ to one approaching k^{-3} therefore reflects the progressive shift from large-scale billow-dominated dynamics to a regime increasingly influenced by fine-scale vortical structures and enstrophy transfer.

V. CONCLUSIONS

A parameter-conditioned multiscale CNN-AE + LSTM framework was developed for the reduced-order prediction of stratified KH instability. The proposed approach uses a State CNN-AE to represent the dominant flow structures and a Difference CNN-AE to capture the finer-scale motions. These representations are then combined and evolved in time using a parameter-conditioned LSTM network, allowing the model to account for variations in the governing flow parameters, J and R.

The framework was evaluated using high-fidelity LES data for both training cases and previously unseen testing cases. The results show that both the traditional CNN-AE + LSTM framework and the proposed approach accurately reproduce the early linear growth of the instability. However, clear differences emerge during the nonlinear stages of evolution. As the KH billows roll up and generate increasingly complex flow structures, the traditional framework gradually loses accuracy, particularly in its representation of the kinetic energy spectrum. In contrast, the proposed framework maintains closer agreement with the LES results and more accurately captures the evolution of perturbation enstrophy and the distribution of energy across spatial scales.

The improved performance of the proposed framework is

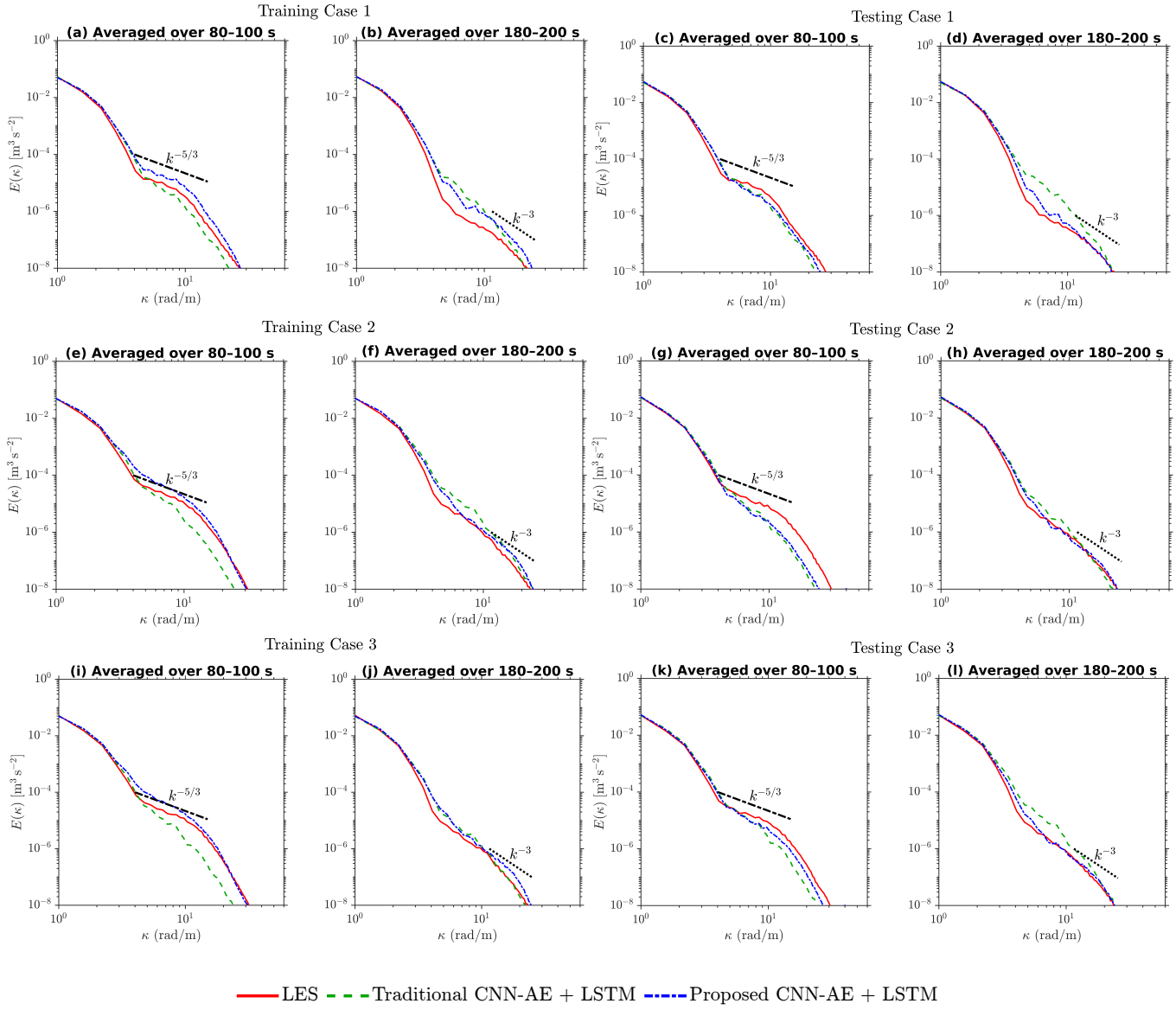


FIG. 13. Comparison of the time-averaged 2D kinetic energy spectra predicted by LES, the traditional CNN-AE + LSTM framework, and the proposed multiscale CNN-AE + LSTM framework for the representative training and testing cases listed in Table V. Panels (a) and (b), (e) and (f), and (i) and (j) correspond to Training Cases 1–3, respectively, while panels (c) and (d), (g) and (h), and (k) and (l) correspond to Testing Cases 1–3, respectively. For each case, the left panel presents the spectrum averaged over the early nonlinear stage (80–100 s), whereas the right panel presents the spectrum averaged over the later nonlinear stage (180–200 s). The dashed and dash-dotted black lines indicate the theoretical $k^{-5/3}$ and k^{-3} scaling laws, respectively. Both frameworks accurately reproduce the large-scale energy-containing motions at low wavenumbers. However, the proposed multiscale CNN-AE + LSTM framework more closely follows the LES spectra over the intermediate and high wavenumber ranges, particularly during the later nonlinear stage, indicating an improved representation of fine-scale flow structures and nonlinear energy transfer across scales.

attributed to two distinct reasons: (a) separation of large and fine-scale features and (b) inclusion of governing flow parameters J and R, which provide information about the base flow configurations. The improvement becomes particularly evident during the later stages of KH evolution, when nonlinear interactions generate a broad range of dynamically important scales. By combining scale-aware flow representations with parameter-conditioned temporal evolution, the proposed framework more accurately reproduces the evolution of the instability across different flow regimes.

A particularly important result of this study is the ability of the framework to perform long-horizon forecasting from a single early-stage flow snapshot together with the governing flow parameters (J,R). While many previous data-driven reduced-order modeling studies rely on multiple sequential snapshots as input for prediction, the present framework is able to accurately reconstruct the subsequent evolution of the flow using only an early-stage flow field. The model successfully repro-

duces key stages of the instability, including billow roll-up and the development of smaller-scale structures. The framework also maintains good predictive performance for previously unseen (J,R) combinations, demonstrating its ability to generalize across different flow conditions.

The results of the present study show that incorporating scale-dependent flow information within reduced-order models can significantly improve the prediction of stratified shear flows. Based on the representative training and testing cases, the proposed framework achieved an approximately 21% improvement in the mean R^2 compared with the traditional approach. Furthermore, the framework compressed the original flow fields from $128 \times 256 \times 2$ variables to a combined latent representation of size $4 \times 8 \times 48$, corresponding to an approximate 43-fold reduction in dimensionality, while maintaining accurate reconstruction of the flow evolution and preserving the key physical characteristics of the instability. In addition to the improved statistical accuracy, the framework better preserved important physical characteristics of the flow, including the evolution of coherent structures and the distribution of kinetic energy across spatial scales.

The present work focuses on the development and evaluation of the proposed data-driven multiscale reduced-order modeling framework for 2D KH instability. While this formulation captures the primary KH dynamics, it does not account for 3D effects such as secondary instabilities, spanwise structures, and transition to turbulence. It therefore serves as a first step toward generalized multiscale reduced-order models for stratified flows, with future work extending the methodology to 3D configurations.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge financial support from the Prime Minister's Research Fellowship (PMRF) scheme, Ministry of Education, Government of India, PMRF-ID 2503535, which facilitated the procurement of computational resources for this study. Partial support was also provided by the Ministry of Earth Sciences, New Delhi [MoES/PAMC/DOM/21-I/2022 (E-13673)], Naval Research Board (NRB-568/MAR/26-27), and Anusandhan National Research Foundation (ANRF/ARG/2025/000738/ENS).

Appendix A: CFL Sensitivity Analysis

A CFL sensitivity analysis was performed to determine an appropriate numerical time step for the LES simulations. A training case with $J = 0.15$ and $R = 1$ was selected, and simulations were repeated using fixed time steps ranging from $\Delta t = 0.01$–0.06 s. For each time step, the maximum advective CFL number over the complete simulation was recorded. As shown in Fig. 14, the selected time step of $\Delta t = 0.02\,\mathrm{s}$ gives a maximum CFL number of approximately $\mathrm{CFL}_{\max} = 0.46$. The maximum CFL number increases with increasing time step, with larger time steps approaching and exceeding unity.

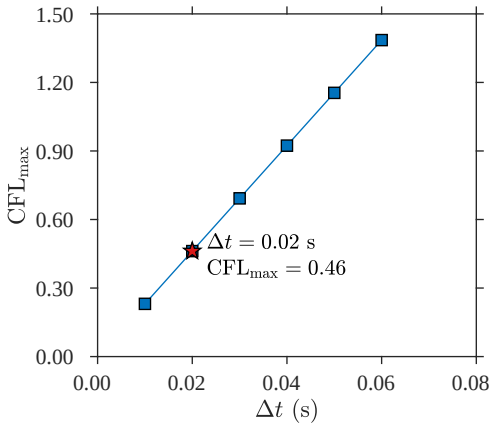


FIG. 14. Maximum advective CFL number obtained for different fixed numerical time steps Δt for the training case with $J = 0.15$ and $R = 1$. The red star indicates the selected time step, $\Delta t = 0.02\,\mathrm{s}$, corresponding to $\mathrm{CFL}_{\max} \approx 0.46$.

TABLE IX. Performance comparison of Network B with 16 and 32 convolutional filters for the training case $J = 0.15$ and $R = 1.0$.

Configuration	t (s)	MSE_Ω	R^2_Ω	MSE_B	R^2_B
16 filters	20	1.340×10^{-4}	0.368	1.000×10^{-6}	0.724
	50	1.496×10^{-3}	0.624	6.000×10^{-6}	0.802
	100	2.369×10^{-2}	0.508	4.300×10^{-5}	0.671
	150	1.480×10^{-2}	0.374	2.600×10^{-5}	0.587
	200	9.197×10^{-3}	0.324	1.400×10^{-5}	0.558
32 filters	20	5.800×10^{-5}	0.728	1.000×10^{-6}	0.851
	50	7.010×10^{-4}	0.824	2.000×10^{-6}	0.934
	100	1.048×10^{-2}	0.782	2.200×10^{-5}	0.830
	150	1.360×10^{-2}	0.425	2.100×10^{-5}	0.662
	200	4.113×10^{-3}	0.697	8.000×10^{-6}	0.769

Therefore, $\Delta t = 0.02\,\mathrm{s}$ provides a sufficient numerical stability margin while maintaining computational efficiency.

Appendix B: Ablation study on increasing the number of convolutional filters in Network B

An ablation study was performed to examine the effect of the number of convolutional filters on the performance of Network B. Two configurations, consisting of 16 and 32 convolutional filters, were evaluated while keeping all other network and training parameters unchanged. The comparison was performed for the training case $J = 0.15$ and $R = 1.0$ at $t = 20$, 50, 100, 150, and 200 s. Table IX presents the reconstruction performance evaluated using MSE and R^2 for both Ω and B. From this snapshot comparison, increasing the number of convolutional filters from 16 to 32 improves the reconstruction performance of Network B for both Ω and B, as reflected by the generally lower MSE and higher R^2 values at the evaluated time snapshots. Therefore, the 32-filter configuration is adopted for Network B in the proposed framework.

Appendix C: Ablation Study of the Output Fully Connected Layer in the LSTM model

To assess the contribution of the output fully connected layer (FCL), an ablation study was conducted by training both the traditional LSTM model and the proposed multiscale framework with and without the output FCL while keeping all other network architectures, hyperparameters, and training procedures unchanged. Since both models exhibited similar trends, only the results for the proposed multiscale framework are presented for brevity. In the model without the output FCL, the output of the final LSTM layer was directly reshaped and passed to the decoders. In the proposed architecture, the output FCL learns the mapping between the LSTM output and the decoder latent space before reconstruction.

Fig. 15 compares the reconstructed flow fields and the corresponding absolute prediction error fields for the representative Training Case 2. Both architectures accurately capture the early evolution of the KH instability. However, even at $t = 50$ s, the model without the output FCL exhibits slight smearing of the Ω and B fields, together with larger local reconstruction errors. These discrepancies become increasingly pronounced during the nonlinear stage ($t = 100$–200 s), where the model without the output FCL progressively loses fine-scale structures and accumulates larger prediction errors. In contrast, the proposed architecture with the output FCL preserves both the large-scale coherent structures and the fine-scale flow features throughout the temporal evolution.

TABLE X. Comparison of the overall prediction performance of the proposed multiscale framework with and without the output FCL for the representative Training Case 2. The reported metrics are averaged over all temporal snapshots and evaluated over the combined Ω and B flow fields.

Model	MSE	R^2
Without output FCL	9.10×10^{-3}	0.83
With output FCL	7.13×10^{-3}	0.86

The qualitative improvements are consistent with the quantitative results summarized in Table X, where the inclusion of the output FCL reduces the mean MSE from 9.099×10^{-3} to 7.131×10^{-3} and increases the mean R^2 from 0.8278 to 0.8598.

These results show that, although the LSTM output and latent-space dimensions are identical (1536), the output FCL provides a more effective mapping between the LSTM hidden representation and the decoder latent space, resulting in improved multiscale flow predictions. Consequently, the output FCL is retained in the final architectures of both the traditional LSTM model and the proposed Network C.

REFERENCES

[1] J. R. Carpenter, E. W. Tedford, E. Heifetz, and G. A. Lawrence, "Instability in stratified shear flow: Review of a physical interpretation based on interacting waves," Applied Mechanics Reviews **64**, 060801 (2011).

[2] A. Guha and G. A. Lawrence, "A wave interaction approach to studying non-modal homogeneous and stratified shear instabilities," Journal of Fluid Mechanics **755**, 336–364 (2014).

[3] S. Thorpe, "Experiments on instability and turbulence in a stratified shear flow," Journal of Fluid Mechanics **61**, 731–751 (1973).

[4] G. Klaassen and W. Peltier, "The influence of stratification on secondary instability in free shear layers," Journal of Fluid Mechanics **227**, 71–106 (1991).

[5] W. D. Smyth and J. N. Moum, "Ocean mixing by Kelvin-Helmholtz instability," Oceanography **25**, 140–149 (2012).

[6] A. Jagannathan, K. B. Winters, and L. Armi, "Stability of stratified downslope flows with an overlying stagnant isolating layer," Journal of Fluid Mechanics **810**, 392–411 (2017).

[7] A. Mashayek and W. Peltier, "The 'Zoo'of secondary instabilities precursory to stratified shear flow transition. Part 1 Shear aligned convection, pairing, and braid instabilities," Journal of Fluid Mechanics **708**, 5–44 (2012).

[8] A. J. Yang, M.-L. Timmermans, and G. A. Lawrence, "Asymmetric Kelvin-Helmholtz instabilities in stratified shear flows," Physical Review Fluids **9**, 014501 (2024).

[9] X. Zhou, J. R. Taylor, and P. C. Colm-cille, "A Lagrangian view of mixing in stratified shear flows," Journal of Fluid Mechanics **1021**, A31 (2025).

[10] C. Caulfield, "Layering, instabilities, and mixing in turbulent stratified flows," Annual Review of Fluid Mechanics **53**, 113–145 (2021).

[11] P. Moin and K. Mahesh, "Direct Numerical Simulation: A tool in turbulence research," Annual Review of Fluid Mechanics **30**, 539–578 (1998).

[12] H. Salehipour and W. R. Peltier, "Deep learning of mixing by two 'atoms' of stratified turbulence," Journal of Fluid Mechanics **861**, R4 (2019).

[13] J. W. Miles, "On the stability of heterogeneous shear flows," Journal of Fluid Mechanics **10**, 496–508 (1961).

[14] L. N. Howard, "Note on a paper of John W. Miles," Journal of Fluid Mechanics **10**, 509–512 (1961).

[15] S. Pawar, S. E. Ahmed, O. San, and A. Rasheed, "Data-driven recovery of hidden physics in reduced order modeling of fluid flows," Physics of Fluids **32** (2020).

[16] J. Wu, Y. Wu, X. Li, and G. Zhang, "Physics-informed neural networks for Kelvin–Helmholtz instability with spatiotemporal and magnitude multiscale," Physics of Fluids **37** (2025).

[17] S. A. Faroughi, N. Pawar, C. Fernandes, M. Raissi, S. Das, N. K. Kalantari, and S. K. Mahjour, "Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks and Operators in Scientific Computing: Fluid and Solid Mechanics," arXiv preprint arXiv:2211.07377 (2022).

[18] P. Sharma, W. T. Chung, B. Akoush, and M. Ihme, "A Review of Physics-Informed Machine Learning in Fluid Mechanics," Energies **16**, 2343 (2023).

[19] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-Informed Neural Networks (PINNs) for Fluid Mechanics: A Review," Acta Mechanica Sinica **37**, 1727–1738 (2021).

[20] A. Hemmasian and A. Barati Farimani, "Reduced-order modeling of fluid flows with transformers," Physics of Fluids **35** (2023).

[21] R. Maulik, B. Lusch, and P. Balaprakash, "Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders," Physics of Fluids **33** (2021).

[22] T. Murata, K. Fukami, and K. Fukagata, "Nonlinear mode decomposition with Convolutional Neural Networks for fluid dynamics," Journal of Fluid Mechanics **882**, A13 (2020).

[23] K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata, "CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different reynolds numbers," Fluid Dynamics Research **52**, 065501 (2020).

[24] T. Bolton and L. Zanna, "Applications of deep learning to ocean data inference and subgrid parameterization," Journal of Advances in Modeling Earth Systems **11**, 376–399 (2019).

[25] R. Halder, M. Ataei, H. Salehipour, K. Fidkowski, and K. Maki, "Reduced-order modeling of unsteady fluid flow using neural network ensembles," Physics of Fluids **36** (2024).

[26] D. Kivarkis, W. Mouhali, S. Benkadda, and K. Schneider, "Autoregressive prediction of 2D MHD dynamics inferred from deep learning modeling," arXiv preprint arXiv:2604.18221 (2026).

[27] K. Fukami, K. Fukagata, and K. Taira, "Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows," Journal of

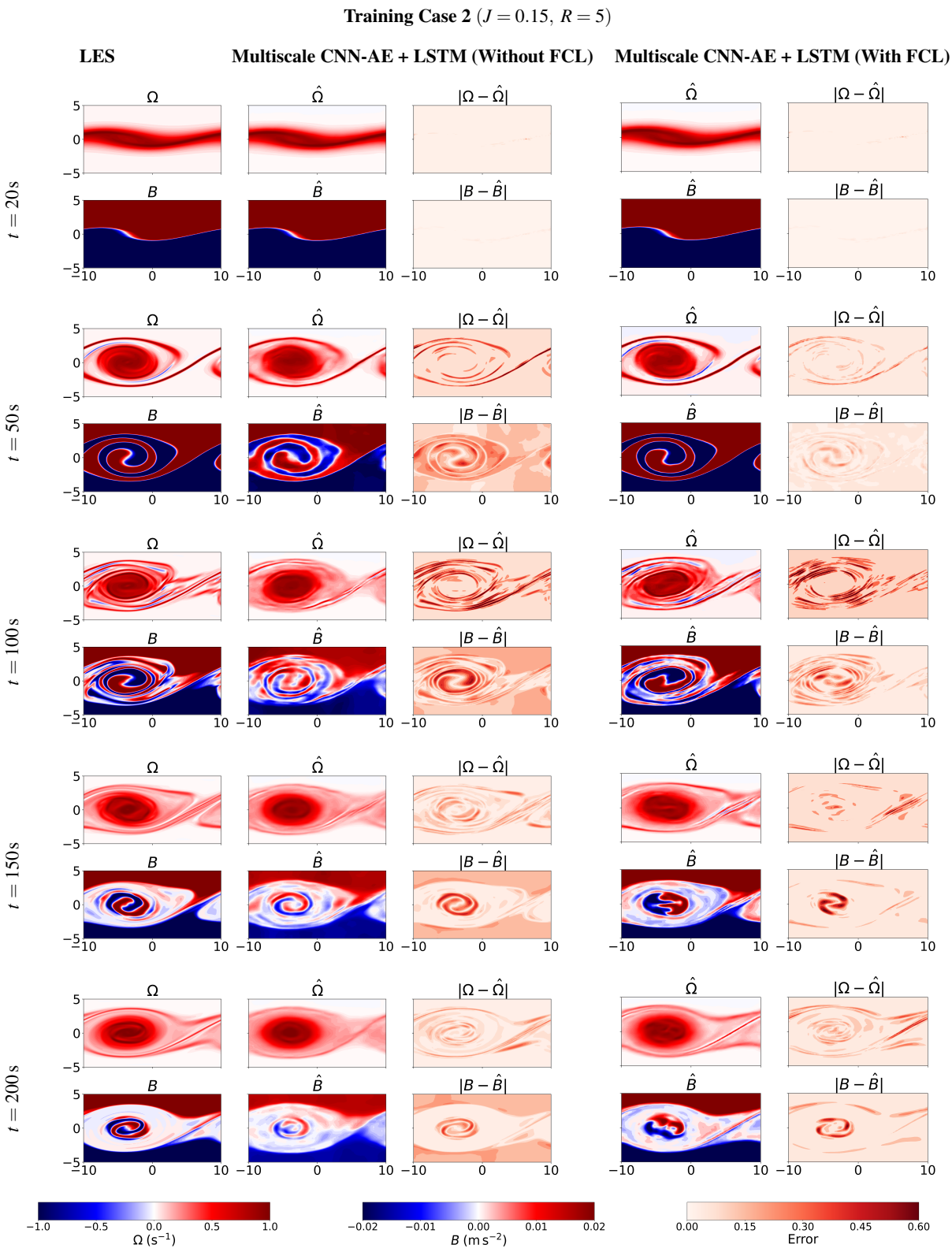


FIG. 15. Same as Fig. 9, but comparing the proposed multiscale framework without and with the output FCL in Network C.

Fluid Mechanics **909**, A9 (2021).
[28] J. Morton, A. Jameson, and M. J. Kochenderfer, "Parameter-conditioned generative modeling of fluid flows," AIAA Journal **57**, 2825–2836 (2019).
[29] G. Klaassen and W. Peltier, "The onset of turbulence in finite-amplitude kelvin–helmholtz billows," Journal of Fluid Mechanics **155**, 1–35 (1985).
[30] A. Ramadhan, G. Wagner, C. Hill, J.-M. Campin, V. Churavy, T. Besard, A. Souza, A. Edelman, R. Ferrari, and J. Marshall, "Oceananigans.jl: Fast and friendly geophysical fluid dynamics on GPUs," Journal of Open Source Software **5**, 2018 (2020).
[31] J. Smagorinsky, "On the numerical integration of the primitive equations of motion for baroclinic flow in a closed region," Monthly Weather Review **86**, 457–466 (1958).
[32] J. Smagorinsky, "General circulation experiments with the primitive equations: I. the basic experiment," Monthly weather review **91**, 99–164 (1963).
[33] D. K. Lilly, "On the numerical simulation of buoyant convection," Tellus **14**, 148–172 (1962).
[34] D. Lilly, "The representation of small-scale turbulence in numerical simulation experiments," NCAR Report (1966).
[35] W. Smyth and W. Peltier, "The transition between Kelvin–Helmholtz and Holmboe instability: An investigation of the overreflection hypothesis," Journal of Atmospheric Sciences **46**, 3698–3720 (1989).
[36] W. D. Smyth and K. B. Winters, "Turbulence and mixing in Holmboe waves," J. Phys. Oceanogr. **33**, 694–711 (2003).
[37] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1 (MIT Press Cambridge, 2016).
[38] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," CoRR **abs/1412.6980** (2014).
[39] A. Jagannathan, K. Srinivasan, P. Damien, M. J. Molemaker, R. Barkan, N. Patnaik, J. C. McWilliams, M. Mathur, and D. Sengupta, "Pathways of forward energy cascades in the northern bay of bengal," Journal of Physical Oceanography **56**, 1215–1243 (2026).
[40] D. C. Fritts, L. Wang, T. Lund, and S. Thorpe, "Multi-scale dynamics of Kelvin–Helmholtz instabilities. Part 1. Secondary instabilities and the dynamics of tubes and knots," Journal of Fluid Mechanics **941**, A30 (2022).
[41] R. H. Kraichnan, "Inertial ranges in two-dimensional turbulence," Tech. Rep. (1967).

.